Topographic Engineering Center

US Army Corps
of Engineers®
Engineer Research and
Development Center

Ascender II
Knowledge-Directed Image Understanding for
Site Reconstruction

Allen Hanson, Edward Riseman, and Howard Schultz

March 2001

20010411 139

Preface

The work described in this final report was performed under a series of contracts from DARPA programs designed to provide softcopy photogrammetric workstations to the country's image analysts. The concern was that in an era when digital imaging promised to provide unprecedented views of the world from all directions and across a much larger fraction of the electromagnetic spectrum there were fewer and fewer image analysts available to exploit this wealth of data. Both the DARPA RADIUS and APGD programs allowed a group of researchers from both academia and industry to interact on very interesting scientific problems and a significant step forward was taken. We need to take more steps.

Table of Contents

List of Figures

List of Figures (Continued)

List of Tables

Ascender II: Knowledge-Directed Image Understanding
for Site Reconstruction

Final Report
Covering the period April 1997 - May 1999

Introduction

This report presents final results from the two years of our APGD research effort on
aerial image reconstruction. It is organized into three sections, covering independent yet
synergistic aspects of our work. Briefly, the first section extends the structure of the
Ascender II system to include utility theory as the basis for decision making in Bayesian
nets. It also contains results of recent evaluation and reconstruction efforts on several data
sets as well as results from a knowledge base that was learned by the system. The second
section details a set of algorithms for recovering (rooftop) surface structure from aerial
images. The third section of the report describes our efforts to recover geometric
building structure from SAR and IFSAR data.

One important task in image interpretation is the process of understanding and identifying
segments of an image. In this effort a knowledge-based vision system is being presented,
where the selection of IU algorithms and the fusion of information provided by them is
combined in an efficient way. Knowledge-based vision systems developed so far have
focused on the interpretation problem for a small set of object classes. A major problem
with these systems is that the knowledge base, control mechanism, and knowledge
sources are combined into a single intertwined system and the addition of new knowledge
or change of domain requires a significant effort. In the current work, the knowledge base
and control mechanisms (reasoning subsystem) are independent of the knowledge sources
(visual subsystem). This gives the system the flexibility to add or change knowledge
sources with only minor changes in the reasoning subsystem. The reasoning subsystem is
implemented using a set of Bayesian networks forming a hierarchical structure that
allows an incremental classification of a region, given enough time.

The control of vision algorithms is performed by an independent subsystem that uses
Bayesian networks and utility theory to compute the marginal value of information for
alternative operators and selects the ones with the highest value. We have implemented
and tested this control structure with several datasets of aerial images. The results show
that the knowledge base used by the system can be acquired using standard learning
techniques and that the value-driven approach to the selection of vision algorithms leads
to performance gains. Moreover, the modular system architecture simplifies the addition
of both control knowledge and new vision algorithms.

Useful representations of the data produced from active and passive range sensing
techniques typically require that the 3-dimensional points are segmented into meaningful
surfaces and that erroneous data are removed. An algorithm is developed in the second
part of this report that automatically segments a range image into coherent surfaces and
reconstructs a 3-dimensional model of the scene. The technique is composed of a two-

phase recursive process. First, a set of points is used to index into a set of surfaces representing the differential geometry of a region. Next, the best set of indexed surface models is used as initial estimates for robust surface optimization in order to converge on the model and parameters that most closely describe the data. After the best-fit surface has been determined for a region, an outlier analysis phase searches for substructures that are recursively processed by the algorithm. The algorithm both segments and reconstructs the scene recursively. The technique is demonstrated on two different scenes, both containing significant amounts of noise, a complex "tabletop" scene of several different objects, and an elevation map of several building rooftops of varying types.

The strength of modern vision algorithms lies not in the ability of any individual algorithm to robustly accomplish its task, but rather in the fusion of information from many sources of data to arrive at an interpretation that represents a consensus of the multiple data sources. The final section of this report deals with the recovery of geometric structure from SAR and IFSAR data. The presence of noise, missing data, and poorly understood radar artifacts in such images necessitates the use of robust and context-sensitive techniques. The algorithm presented here exploits knowledge about the geometric structure of buildings and how this geometry interacts with the sensor.

Rooftops are extracted in two stages. In the first, a building's back edge is located by way of the shadow it casts in the image. Once the back edge of a building has been found, its rooftop is extracted through region growing. The region's growth begins at this back edge, and proceeds along the building's boundary. Once growth has terminated, a rectangle is fit to the rooftop region. The initial findings, as presented here, are for buildings with a rectangular boundary, although work is under way for recovering more complex boundary types.

1. Control in a 3D Reconstruction System Using Selective Perception

An Image Understanding (IU) system should be able to identify objects in 2D images and to build 3D relationships between objects in the scene and the viewer. A large number of image understanding systems developed so far are dedicated to Aerial Image Interpretation. One of the problems with Aerial Image Interpretation systems is the management of uncertainty. Uncertainty in this case arises from a variety of sources, such as the type of sensor, weather conditions, illumination conditions, season, random objects in the scene, and the inherent uncertainty in the definition of common objects.

Object recognition in aerial images is one important step towards 3D reconstruction of a scene, but automating the recognition process in a real-world application is not an easy task. Consider the image tiles from aerial images presented in Figure 1.1. The tile on top contains a building, which is easy to identify by its door and rooftop. The recognition of the three objects marked in the bottom tile is not as simple, and more comparisons and measurements may be required to identify them correctly.

Since an interpretation of an image can be viewed as a correspondence between image features and the identifying object classes, it is clear that the descriptive vocabulary of the system must be reflected in the set of features extractable from the image. Thus the image features must form the primitive descriptions of the objects in the knowledge base. Since every feature has at least one operator for measuring it, the control problem addressed in this paper is this: given a general-purpose system and a specific interpretation problem within the domain of the system, how does one effectively select the features to measure or, more generally, which algorithms to apply, and in what order. Furthermore, because there is a significant amount of inherent ambiguity in the interpretation process, an interpretation system must include a sufficiently rich set of relations among features as well as flexible mechanisms for manipulating uncertain hypotheses until there is a convergence of evidence.

In this section, the structure of the Ascender II system is reviewed and how to use Bayesian networks and utility theory to build a control structure for a general-purpose image-understanding system. We also address the knowledge engineering issue by demonstrating that it is possible to learn the Bayesian network structures from fairly coarse training information. Ascender II, an IU system for fully automated Aerial Image Interpretation, is used as a testbed to address these questions:

- How can the results of visual operators and their associated uncertainty be combined in order to classify a particular region?
- How can the hierarchical structure of objects be exploited in order to construct an incremental classification process?
- Can the construction of the knowledge base be simplified (or fully automated) for a particular application using both human expertise and machine learning techniques?
- How can performance be improved by using a disciplined approach to operator selection?

The next section presents previous work in vision systems. Section 3 introduces the Ascender II system and presents its control structures, specifically how operators are ordered given the current knowledge. Section 4 shows how to learn the structures used for control. Experimental results are presented in Section 5 and conclusions plus future direction of this work are outlined in Section 6.

1.2. Background

One popular approach in the 1980s to the general Image Understanding problem was knowledge-directed vision systems. A typical knowledge-directed approach to image interpretation seeks to identify objects in unconstrained two-dimensional images and to determine the three-dimensional relationships between these objects and the camera by applying object- and domain-specific knowledge to the interpretation problem. A survey of this line of research in computer vision can be found in Haralick and Shapiro 1993, Draper et al. 1996, and Crevier and Lepage 1997.

Typically, a knowledge-based vision system contains a knowledge base, a controller, and knowledge sources (or visual operators). In most of these systems the controller and the vision algorithms are combined into a single system. Some of the problems common to most of the knowledge-directed vision systems are the following: control for vision procedures was never properly addressed as an independent problem [Draper et al. 1996], the system's structure did not facilitate entry of new knowledge [Crevier and Lepage 1997], and the knowledge engineering task was formidable [Draper et al. 1996]. These are some of the issues that are addressed in this paper.

Bayesian networks have been successfully used in systems required to combine and propagate evidence for and against a particular hypothesis. Vision systems have been developed using Bayesian networks for knowledge representation and as a basis for information integration, e.g., Rimey and Brown 1992, Mann and Binford 1992 and Krebs et al. 1998 (for indoor applications), and Kumar and Desai 1996 (for aerial image interpretation).

1.3. Value-Driven Control of a Vision Algorithm

The Ascender II system was designed for aerial image interpretation, particularly for the 3D reconstruction of urban areas. The system is divided into two independent parts — the reasoning subsystem and the visual subsystem — running on different operating systems on different machines, as shown in Figure 1.2. One advantage of this design is that changes to the reasoning subsystem, or to the visual subsystem, can be made independently.

Although the initial effort has focused primarily on recognizing and reconstructing buildings from aerial images, Ascender II has been designed as a general purpose vision system. The system has a set of focus-of-attention regions as input. These regions can be extracted from aerial images automatically (using a system such as Ascender I [Collins et

al. 1998]), manually, or interactively (using cues from other sources such as maps or other classified images). The system's goal is to select vision algorithms, recognize objects in the scene, and reconstruct these objects in 3D automatically.

The system's knowledge base is composed of a set of Bayesian networks organized hierarchically. The Bayesian networks are used to integrate information from different sources, and to label a region based on information provided by visual operators. Each level of the hierarchy represents object classes at a specific scale [Jaynes et al. 1998a,b]. The hierarchy leads to a system capable of performing incremental classification. The classification process is refined until the hierarchy reaches its finest level, or until the system exhausts all resources available. The Bayesian networks were developed using the HUGIN system [Jensen 1996].

The first set of networks was developed manually; two of the five networks used in the system are presented in Figure 1.3 and 1.4. The root node corresponds to the region of discourse at a specific level of detail. All leaf nodes correspond to visual operators, and all internal nodes correspond to features that can be measured in the image. The probability table associated with the links between a feature node and an operator node reflects the reliability of the operator in retrieving the value of the feature; a link between the root node and the internal nodes represents relationships between object classes and feature values. The probability tables related to these links reflect the probability that a feature has a certain value given that the region is a certain object class, or

$$P(\text{Feature}=k \mid \text{Region}=\text{Object}_i).$$

A set of experiments has been performed to compare alternative evaluation measures for operator selection. The first of these, called uncertainty distance [Marengoni et al. 1999], represents the difference between the value of the maximum belief in a node and the value of the belief if the node had a uniform distribution. Given a network, the system computes the uncertainty distance for all nodes that have a correspondent IU process and selects the node with the minimum uncertainty distance. This was shown empirically to be equivalent to entropy as an evaluation measure [Marengoni et al. 1998].

The performance in terms of classification of the system using uncertainty distance when compared with a system that used all available information was about the same, but the system using uncertainty distance used a smaller number of operators [Marengoni et al. 1999].

The work presented here uses the same system architecture, but it employs a more sophisticated measure to select visual operators, namely utility theory [Lindley 1985]. Utility theory is a probabilistic technique for decision making and it fits well in a Bayesian network system. Utility theory selects the decision that has the highest expected utility. In the discussion that follows, the following notations are used:

- R_j $\stackrel{\text{def}}{=}$ region R belongs to Class j.

- DR_j $\stackrel{\text{def}}{=}$ The decision that region R is identified as Class j.

- E_j $\stackrel{\text{def}}{=}$ All the evident collected so far.

- F_m $\stackrel{\text{def}}{=}$ Feature F has m discrete states.

The expected utility (EU) of each decision is computed using the probability that a region belongs to a class j, $P(R_j|E)$, and the utility of deciding that a region is in Class i given that the region belongs to class j, $U(DR_i| R_j)$ [Lindley 1985]:

$$EU(DR_i|E) = \sum_{j=1}^{N} U(DR_i|R_j) * P(R_j)$$

The current utility of the decision is defined as the maximum value among each of the expected utilities:

$$\max_i (EU(DR_i|E))$$

The best decision is defined as the decision _ which gives the maximum expected utility:

$$\alpha = \text{argmax}_i (EU(DR_i|E))$$

In our problem domain the system has to decide the most likely identity (e.g. label) of a region. Assume that there are K features that can be measured in the region, the measurements are not completely reliable, and the measurements help in deciding about the region's label.

The prior probabilities about the region's label and the conditional probabilities relating features with labels are stored in the Bayesian networks. The utility tables storing the values $U(DR_i|R_j)$ are not hard to define. The utilities represent a personal desire for the system behavior, in this case only the correct labels are accepted. The utility tables are all similar, with ones in the diagonal and zeros in all other entries (see Table 1.1). The utility

Table 1.1. Utility values $U(DR_i|R_j)$ for the level 0 network in Ascender II.

	Building	Park. Lot	Open Field	Other
Building	1	0	0	0
Parking Lot	0	1	0	0
Open Field	0	0	1	0
Other	0	0	0	1

values in this table can be adjusted by the user of the system to reflect his/her desire in the classification process [Lindley 1985].

Features are selected based on the value of information [Howard and Matheson 1984] associated with each feature. This value is computed as follows: for each feature available, compute the expected utility of the system given that information about the feature is known.

$$EU(DR_i \mid E, F_m) = \sum_M P(F_m) * \max_i(EU(DR_i \mid E, F_m)$$

Now, compute the value of information of each feature as follows:

$$VI(F_m) = EU(DR_{\alpha'} \mid E, F_m) - EU(DR_\alpha \mid E)$$

and select the feature with the highest value of information. Intuitively, the value of information measures the expected improvement in the utility of the best decision, once the result of an operator becomes available.

Figure 1.5 shows a generic Bayesian network that will be used to illustrate how feature selection is performed in the Ascender II system. The first step is to compute the system's utility before extracting any information about the features. Each decision has an expected utility $U(Dec_i) = EU(DR_i\} \mid E)$; the expected utilities of the decisions can be calculated by multiplying the matrix of utilities by the column vector of beliefs from the root node, as shown in Figure 1.5. The system's utility is the maximum value among the utilities of the decisions.

The next step is to compute the value of information of each feature. This is performed by computing the expected utility of each feature as follows: assume feature "i" has "M" states, $state_1$, $state_2$, ..., $state_M$; each state in feature "i" has a corresponding belief, bel_1, bel_2, ..., bel_M. These beliefs correspond to the current expectation about the outcome of feature "i." Set the outcome of feature "i" to $state_1$ (make the belief of $state_1=1$ and the belief of all other states equal to 0), and propagate the information through the network. This will change the beliefs in the states of the root node. Use this new set of beliefs in the root node to compute the new utility of the system. When completed, the value of information is found from equation 1.

1.4. Learning the Models for the Control Structure

The knowledge engineering necessary to design an efficient Bayesian network (structure and probability tables) is a time-consuming task, even for small networks such as those currently used in the Ascender II system. This has been one of the main criticisms of Bayesian networks.

Algorithms for learning Bayesian networks from data have been developed [Breese and Heckerman 1995; Cheng et al. 1997]. Cheng's algorithms [Cheng et al. 1998] are based

on statistical measures over the random variables, computing correlation between two variables using mutual information, and conditional mutual information given a third variable, to define causality. Cheng's algorithms were used to learn the structure and the probability tables for the networks in the Ascender II system.

The data used for learning were collected from three different well-known data sets (Fort Hood, Fort Benning, and Avenches); overall, 79 regions were selected representing a mix of objects drawn from buildings, parking lots, grassy fields, etc. All regions were presented to a set of six human subjects, and the subjects were asked to estimate the state of each feature in the feature set (each feature value was coarsely discretized to facilitate the human task). This information was compiled and used to learn a Bayesian network representing the task domain.

Note that the structures as learned contain only the node representing the region plus the nodes representing all the features. The operator nodes were added manually after the learning phase along with their reliability tables. If the true value of each feature is known, the tables representing the operator's reliability can also be learned from the data.

The learned networks corresponding to Figures 1.3 and 1.4 are shown in Figure 1.6 and 1.7. The general structure is completely different, although some of the substructures were preserved. Also, the learned networks are generally more densely connected.

The networks learned from data are limited to the objects present in the training data. For instance, the data used to learn the networks had only peak- and flat-roofed buildings. Thus the feature *Rooftop* in Figure 1.7 has only states for *Peak* and *Flat* roofs, and not the more general structure as in the handcrafted networks presented in Figure 1.4.

1.5. Results

A set of experiments was performed on the Fort Hood data set (seven views with known camera parameters and corresponding digital elevation map DEM) shown in Figure 1.8a, on the Avenches data set (one view and a DEM) shown in Figure 1.8b, on the Fort Benning data set (two views and a DEM) shown in Figure 1.8c, and on the ISPRS Flat data set (two views and a DEM) shown in Figure 1.8d. These data sets are an effective test suite because they have different numbers of images, different resolutions and different numbers of objects in each class.

The first experiment was designed to show that a more disciplined approach to feature selection leads to a more efficient system. The experiment provides a comparison between the system using uncertainty distance (Basic System) and the system using utility theory (System A). Both systems used the handcrafted networks. The results in terms of classification and number of operators used are presented in Tables 1.2 and 1.3.

Table 1.2 shows that the overall classification obtained by the two selection processes is about the same. Table 1.3 shows that the selection of operators is more efficient using utility theory (10% fewer operators). This result confirms the intuition that a selection

methodology using utility theory would choose more effective operators, thus classifying regions faster.

Table 1.2. Summary of the recognition process for different data sets using the handcrafted networks. In each case the number of objects correctly identified is shown, followed by the total number of objects evaluated by the system.				
Uncertainty Distance: Basic System				
Data Set	Overall	Level 0	Level 1	Level 2
Fort Hood	34/42	36/42	22/24	21/21
Avenches	12/18	15/18	12/13	5/7
Fort Benning	17/19	18/19	17/18	17/18
Utility Theory: System A				
Data Set	Overall	Level 0	Level 1	Level 2
Fort Hood	35/42	37/42	23/25	21/21
Avenches	13/18	16/18	12/13	5/7
Fort Benning	16/19	18/19	17/18	16/17

Table 1.3. Total number of calls to visual operators for all data sets for all classes.	
Decision Process	*Number of Operators*
Utility Theory	430
Uncertainty Distance	475

The second set of experiments was designed to demonstrate the performance of the system using the learned networks on the same data sets used for training. Although the regions used in these experiments are the same as the ones used for learning, there are two major differences that have to be considered:

1. During the experimental phase the features were computed algorithmically from the image data by a visual operator. The results do not necessarily correspond to the outcome given by humans in the learning phase.

2. The values of the features computed by the visual operator were entered into the operator's node and were attenuated by the operator's reliability during the propagation.

First, the networks and probability tables (including prior probabilities) as learned from the data (System B) were applied in the three data sets (Fort Hood, Avenches, and Fort Benning). Because the prior probabilities learned from data reflect the exact frequency of each object class, the system should react faster to feature values retrieved and it would not be a fair comparison to System A. So, a second test was performed where the prior beliefs for each object class were changed in the networks to reflect the same prior probabilities used in the handcrafted networks (System C). The results obtained for these two experiments are shown in Tables 1.4 and 1.5.

Table 1.4. Summary of the recognition process for different data sets using the learned networks. In each case the number of objects correctly identified is shown, followed by the total number of objects evaluated by the system.

Learned Networks: System B

Data Set	Overall	Level 0	Level 1	Level 2
Fort Hood	33/42	34/42	20/21	20/20
Avenches	16/18	18/18	15/15	7/9
Fort Benning	15/19	18/19	17/18	15/17

Learned Networks + Modified Priors: System C

Data Set	Overall	Level 0	Level 1	Level 2
Fort Hood	34/42	35/42	20/21	20/20
Avenches	13/18	16/18	12/14	6/7
Fort Benning	16/19	18/19	17/18	16/17

Table 1.5. Total number of calls to visual operators for all data sets for all classes.

Decision Process	*Number of Operators*
Learned Networks	322
Learned + Modified Priors	400

The numbers shown in Table 1.4 are similar to the numbers presented in Table 1.2. Thus, the system using Bayesian networks learned from data generates classifications very similar to the system using the handcrafted networks. However, System B was able to classify the regions using 32% fewer operators than the Basic System. System C used 15% fewer operators than the Basic System. The fact that System C used more operators than System B was expected because the distributions of beliefs over the object classes were more uniformly distributed in System C than in System B, thus System C requires more exploratory calls before deciding about a region.

The third experiment was designed to show that the structure and relationships among features learned from data are robust enough to be applied to a different data set. In this experiment, the handcrafted system using utility theory was compared to the learned system applied using the Flat data set. In both systems the prior beliefs were adjusted accordingly. The results over 30 regions are shown in Table 1.6.

Table 1.6. Summary of the recognition process for the Flat data sets using the handcrafted and the learned networks with utility theory.

Flat Data Set

System	Overall	Level 0	Level 1	Level 2	Operators
Handcrafted	22/30	23/30	21/21	13/14	170
Learned	16/18	18/18	15/15	7/9	160

The number of operators used by the system using the learned networks is slightly smaller (5%), but the larger number of relationships between the features in the learned networks allowed a better performance of the system in the new data set (87% correct classifications against 73% for the system with the hand-crafted networks).

One example of the 3D reconstruction that can be obtained using the Ascender II system is presented in Figure 1.9. The maximum error between the reconstructed buildings and the CAD models handcrafted for the buildings in the Fort Benning data set is less than 1.2 meters.

1.6. Conclusions and Future Work

The overall performance of the Ascender II system using utility theory or uncertainty distance is above 80% in terms of classification. When utility theory and value of information are used, the system selects operators more efficiently and is able to identify objects faster.

The knowledge base in Ascender II is based on Bayesian networks. Evidence from different sources is combined in the Bayesian networks and each contributes to the region classification. We have also shown that the networks can be learned from data. The system using the learned networks had a better performance, either in terms of the number of operators required to correctly classify the regions, or in terms of the percentage of regions correctly classified. The data used to learn the networks have to be representative of all objects classes desired in the system. The learned networks are robust enough to be applied in a different data set with a simple adjustment of prior beliefs for the object classes.

The hierarchical structure leads to a system capable of performing incremental classification. The current system can be adjusted to behave as an anytime system, where resources, such as number of operators or processing time, can be limited and the overall performance optimized for the resources available.

Another possible extension of this system is related to temporal reasoning. If a 3D reconstruction of a site is available and a new image is obtained for the same area, how can the information previously computed be used to drive the system in order to detect changes and to reconstruct the new site efficiently?

2. Recursive Recovery of Three-Dimensional Scenes

2.1. Introduction

In this section, the problem of both segmenting an unstructured set of range estimates into coherent regions and, for each region, determining the underlying surface are addressed. The typical approach to scene reconstruction has been to view segmentation and reconstruction of the scene as two independent problems or to assume that the entire set of range estimates represents a single surface. As a consequence, there have been

significant advances in the range segmentation problem (see [Besl and Jain 1985]),
particularly through surface growing techniques [Besl and Jain 1988, Taubin 1991, Fua
1995, Miller and Stewart 1997] and the problem of model fitting through a number of
approaches, including deformable models [Kass et al. 1988, Terzopoulos and Metaxas
1990, Cohen et al. 1991], global model estimation and registration [Zhang 1994], and
mixed approaches [Montagnat and Delingette 1997]. Although there has been promising
work in surface reconstruction that makes use of optical images [Fua 1995, Jaynes et al.
1997a,b], and the use of constraints derived from the formation of a range image from a
stereo pair [Fua and Leclerc 1994], the approach we take here assumes that
corresponding greyscale images are unavailable. See [et al. 1997a,b] for similar work
under the assumption that both an optical image and an elevation map are available.

The algorithm proceeds recursively in two phases: model estimation followed by model
verification. Model estimation indexes into a library of parameterized models using a set
of measured 3-dimensional points. The library of models is rank-ordered according to a
similarity measure based on the differential geometry of the points. The model
verification phase uses the set of parameterized models in the library that most closely
matches the measured points as initial estimates for a robust fitting procedure. The model
that converges to the lowest residual fit error is then used to reconstruct the set of points.
Outlier points, with respect to the reconstructed surface, provide a basis for further
segmentation and are clustered into new regions for recursive processing. Regions are
removed from the scene in two ways. Either the region is removed during the outlier
filtering phase on the basis of morphological constraints, or it is eliminated if a robust
fitting fails to provide a sufficiently good solution.

In order for the algorithm to be successful, two conditions must hold: 1) the scene is
composed only of the models in the algorithm's database, and 2) at any one phase of the
recursive process, more than half of the points of a region must lie within one of these
models. The first requirement is fairly straightforward: the model-directed nature of the
problem assumes that models for estimation and reconstruction are available.
Requirement two is common to most robust fitting techniques. Both the model-indexing
scheme and the final surface fit require that at least half of the range measurements
within the region under consideration arise from a single model.

The algorithm may be particularly useful in robotics, for example, to determine both the
location and the differential properties of objects for grasping. The recursive nature of the
algorithm has applications in high-resolution cartography, where complex building
rooftops, containing substructures such as dormers, chimneys, and spires, can be
automatically reconstructed.

In order to demonstrate the performance of the algorithm, a range image was generated
from a CAD model of a scene containing seven different "Lego" blocks and two pencils
resting on a tabletop surface. Range estimates were produced using a synthetic range
sensor, placed directly above the scene, oriented nadir to the table surface. The three-
dimensional points generated from the sensor were then modified using Gaussian noise
with $\mu=0.0$ and a σ value of 0.2. This was followed by the introduction of noise in which

10% of the (x, y, z) points were randomly perturbed with a Z-value error with a standard deviation of 2.65 centimeters (1.5 times that of the maximum Z in the scene) and an XY error with a standard deviation of 0.3 centimeters.

The goal of the algorithm is to simultaneously segment each of the objects from the background and to reconstruct their geometry and corresponding substructures. Figure 2.1 shows the rendered model of the scene and a close-up of the 3-dimensional points acquired from a synthetic range sensor placed directly above the scene.

2.2. Model Estimation through Indexing

The estimation scheme indexes into a library of surface primitives based on an analysis of the differential geometry within a region of the range image. The estimated orientations of small patches are used to construct a Gaussian image [do Carmo 1976, Horn 1986, Zhang 1997] that is then correlated with the model library. Correlation provides an orientation vector, and a rotation about that vector at which the histograms correlate maximally. The set of models used for the results in this paper is shown in Figure 2.2. The model library contains seven model classes, and 42 models representing the various possible parameterizations.

Model parameters describe aspects of the model shape itself. For example, the *Peak* model is represented by a distance along a center axis and the angle between the two planes. The number of parameters for each surface in the library is shown at the left of each model in Figure 2.2.

The set of points $x^P = (x, y, z)_k$, within a region k of the range image, are triangulated into a simple surface using the Delauney algorithm [Aurenhammer 1991]. If no regions are available, as is the case initially, then all points in the range image are used. The computed surface mesh is a set of triangular patches, $T_i = (p_1, p_2, p_3)$, where p_1, p_2, and p_3 are points from the range samples. x_i^P is defined as the point equidistant from p_1, p_2, and p_3 for triangle i. The surface normal for each patch at x_i^P is then computed as

$$\overline{N_{x_i^P}} = \frac{(p_2 - p_1)}{\|(p_2 - p_1)\|} \times \frac{(p_3 - p_1)}{\|(p_3 - p_1)\|} \tag{1}$$

It is assumed that the vector representing the surface normal pointing "out" from the scene (as opposed to the vector pointing towards the center of objects) is known. This surface normal is used to determine the cell on the Gaussian sphere that will receive a "vote" for a particular orientation.

To avoid sensitivity problems with the method by which the orientation space is subdivided into discrete regions on the sphere, votes are smoothed using a Gaussian distribution. If the surface normal $N_{x_i^P}$ intersects the sphere at $(x, y, z)_s$, the weighted vote is given by

$$V((x, y, z)_i, B) = c_i \frac{1 \cdot}{\sqrt{2\pi\sigma^2}} e^{-(D^2/\sigma^2)} \tag{2}$$

where D is the angular distance from $(x,y,z)_s$ to the center of the histogram bucket, B, to receive the weighted vote and c_i represents the area of surface patch i that is contributing to the vote. The amount of smoothing is related to the expected noise in the range image. However, as σ increases, the separability of the model classes degrades. For the results shown here, $\sigma = 0.3$ and the orientation histogram contains 240 buckets, reflecting a tessellation based on the semi-regular icosahedron [Horn 1986].

A single surface normal may induce a smoothed vote over several buckets, as shown by equation 2, and votes for a given vector no longer contribute when the bucket value of $V((x,y,z)_i, B)$ falls below a threshold (0.1 for the results shown here). Figure 2.3 shows a single region of the "toys" scene (see Figure 2.7b for region labels) after a surface mesh has been fit along with the computed histogram.

To achieve model indexing, the constructed Gaussian image, referred to as the image histogram, is then correlated with each of the model histograms stored in the library. The normalized cross-correlation score is given by

$$C_{\theta, \overline{O}}(I,M) = \frac{\sum^{(i,j)}(I(i,j) - \mu_I)(M(i,j) - \mu_I)}{(\sigma_I * \sigma_M)} \tag{3}$$

where μ and σ represent the mean and variance, respectively, of each of the image and model histograms.

To select the correct relative orientation of the image histogram and the model histogram, the value of $C_{\theta,\overline{O}}(I,M)$ must be computed for many possible values of around several different axes of rotation given by $\overline{O}$. Each of these axes and angles reflects a different alignment between the Gaussian images of I and M. Prior knowledge about the scene domain (that rooftop models align with the gravity vector, for example) can be used to reduce the number of different values of $\overline{O}$ and θ. For the results shown on the tabletop scene, the gravity vector was aligned with the Z-axis and $\overline{O}$ was restricted to 49 different orientation vectors within 30 degrees of the Z axis above the horizontal plane, and θ was restricted to single degree increments about each axis. This allows each model in the database 17,640 different relative orientations between the library model and the extracted histogram.

Each of the models in the library was correlated with the histogram shown in Figure 2.3b. The maximum correlation and orientation parameters for the best three models are shown in Table 2.1. The **Peak,35** model correlated maximally with region 4. Figure 2.4a shows the maximum correlation response for the **Peak,35** model about the 49 different orientation vectors. Figure 2.4b shows the correlation scores for the different values of θ through 360 degrees about $\overline{O}$; the maximum correlation was found at $\theta = 2.09$.

Table 2.1. Top three models matched to the region shown in Figure 2.3a. No value for θ is reported for the plane model because it is circularly symmetric. All three models are fit to the region to determine the appropriate reconstruction.

Model Name	Correlation	$\overline{O}$	Rotation Angle θ
Peak,35	0.836	(0.38, 0.0, 0.92)	2.09
Peak,25	0.797	(0.13, 0.0, 0.99)	2.13
Plane	0.664	(0.18, 0.18, 0.97)	

2.3. Model Verification

Model indexing provides an ordering over the set of models $M_i(x;\overline{a})$ and associated parameters within the model library for a set of points within a region of the range data x^P. The parameter vector $\overline{a}$ and the model M are used as initial estimates for a robust surface fitting procedure. The top several models are fit to the data points and the model that converges to the best fit is used to interpret the data.

Surface fitting involves a multidimensional optimization scheme for $M(x^P, \overline{a}) = 0$ where $\overline{a}$ is the parameter vector associated with the model being fit. Because a triangular mesh has already been fitted to the range data, the surface normal at each patch $\overline{N}_{x_i^P}$ is used to compute the distance between the current model and the observed data. Specifically, the median of

$$E_P[|\hat{x}^P - x^P|^2] \qquad (4)$$

is minimized, where

$$\overline{x^P \hat{x}^P} \mid \hat{x}^P = t \cdot \overline{N}_x P + x^P, \quad \hat{x}^P \in M(\overline{x};\overline{a})$$

that is, $\hat{x}^P$ is the point on the fit surface corresponding to x^P and is obtained along the computed surface normal. This median squared error function avoids measuring error in an arbitrary way, and uses information from the surface mesh to estimate an appropriate direction from the observed data to the model surface. This is particularly important in the case of models with sharp surface discontinuities (the peak model, for example), where error measured near the peak and along the Z-axis may induce an unusually large error.

A multidimensional simplex method [Nelder and Mead 1965] is used to minimize equation 4 over the *k*-dimensional space induced by the number of free parameters in the selected model. In order to avoid optimization over a large number of parameters, neither position nor absolute rotation is part of $\overline{a}$. Note that absolute rotation is computed as part of model indexing from the computation of θ and $\overline{O}$ as the vector at which there is a maximum correlation response between the two histograms. Absolute position in the scene is fixed as the center of the region of data being fitted. Therefore, models are

restricted to move along $\overline{O}$. For example, the plane model has one free parameter after model indexing — its distance along $\overline{O}$.

Outliers are computed as points in the range data that have a relatively high residual error. Because the outlier measure, with respect to the model $M_i(x;\overline{a})$, is the basis for the segmentation of new surfaces, it is important that outliers are not computed from a simple error-prone threshold on E_p. Instead, outliers are computed on the fly through multiple fits using the simplex method. At each iteration, the points with the largest error measure are discarded as outliers, leaving k inlier points for a new fit using the same procedure.

A χ^2 per degrees of freedom measure[1] is used to determine when discarding outlier points no longer improves the surface fit:

$$\frac{\chi^2}{k-1} \qquad (5)$$

where

$\chi = \sigma(E_p)$
k = number of inlier points.

When the value of equation 5 does not decrease as more outlier points are removed from the data, the process stops. Using this technique, the number of outliers removed at each step can be small and is not dependent on characteristics of the data, as a simple threshold based on E_p would be. Figure 2.5 shows a close-up view of object #4 in the tabletop scene and the reconstructed surface obtained by fitting the **peak,35** model to the data by minimizing the least median error as described above.

2.4. Outlier Clustering

After a surface has been fit to the data using the procedure described in the previous section, data points are classified as either inliers or outliers. Outlier points are then clustered into spatially coherent regions and the algorithm is recursively applied to the extracted regions.

Production of valid outlier regions is a straightforward, three-step morphological process. First, a closing operator creates connected component clusters in the range image. An opening operator removes small sets of residual points due to noise. Finally, a dilation step creates complete connected regions. Each region is discarded based on a size constraint that can be derived from the expected minimal size of objects and the known sensor model.

Figure 2.6 shows the outlier points with respect to the peak model fit to region 4. Outliers are due to noise in the range data, inaccurate model fits, and substructures present in the

---

[1] Originally suggested by Howard Schultz via personal communication.

scene. Although object 4 is curved near the side boundaries of the top face (see Figure 2.5a), the library contains no such surface and the peak model was fit. This produces the long bands of outliers (Figure 2.6a) near the boundaries. Figure 2.6b shows the remaining regions after outlier clustering that are recursively processed by the algorithm.

2.5. Results and Conclusions

The algorithm was run on two different scenes. Because the "tabletop" scene was generated from ground-truth models, it was used to study the accuracy of the algorithm. Another test was run on the Ascona ISPRS, "flat" scene; specifically, the elevation map of several buildings that was produced from a stereo optical routine was used as the input data.

2.5.1 Tabletop Experiment

Figure 2.7a shows the actual range data used to reconstruct the tabletop scene. The image is 512 x 512 pixels with a spatial resolution of 12.36 samples per centimeter. The synthetic range sensor was perpendicular to and located above the table surface. Initially, the algorithm recognized a plane and reconstructed the table surface. All outlier regions, with respect to this fit, were then discovered and clustered. Each of the remaining regions after the algorithm terminated are labeled and shown in Figure 2.7b. For each of the regions shown, new outlier regions may have been produced and reconstructed. These were all correctly detected as planar segments above the objects. Two subregions within region 2 were reconstructed as a single surface. As the number of points within a region becomes small, the clustering algorithm is sensitive to the presence of noise and can merge regions located near one another.

Figure 2.8 shows the reconstructed scene. The scene is a set of recovered surfaces in the world coordinate system. Of course, the hidden surfaces (with respect to the range sensor) are unknown and are not part of the reconstructed scene. Accuracy was tested using three different measures: (1) a distance from the center of mass of each ground-truth model to the center of mass of each acquired model, (2) an orientation error in the (x,y) plane, and (3) a coverage percentage computed in pixels. Table 2.2 shows the errors for each of the nine regions, and the computed total RMS error for all of the subregions.

2.5.2 Building Reconstruction Experiment

The second test was performed in the aerial image domain using an elevation map reconstructed from a downlooking stereo pair of the Ascona/ISPRS, "flat" scene. The scene contains five peaked roof buildings of complex rooftop structure. Because building rooftops are expected to be perpendicular to the gravity vector, relative orientations in the model-indexing phase were restricted to rotations about the Z-axis.

The system was run and a local ground plane was fit, producing 12 initial subregions for further processing. Of the 12 sub-regions, seven remained after processing. Two nonbuilding regions were reconstructed as part of the final scene. A dome with a 1/2

base-to-height ratio was reconstructed at the location of a group of trees (see bright circle, top right of Figure 2.9b). A long row of trees was also reconstructed as a cylinder with a 1/3 base-to-height ratio.

Table 2.2. Results of tabletop reconstruction. Regions 3 and 8 have unusually high overlap error due to the fact that the tips of the pencils were not reconstructed; see text for a description of the errors reported here.

Region	Center of mass distance error	Orientation error in x--y plane	Coverage percentage on a pixel basis
1	0.169 cm.	0.018	99.8
2	0.081 cm.	0.031	96.5
3	0.322 cm.	0.018	82.1
4	0.299 cm.	0.023	89.6
5	0.172 cm.	0.037	98.5
6	0.065 cm.	0.013	99.5
7	0.449 cm.	0.103	90.3
8	0.417 cm.	0.019	81.2
9	0.171 cm.	0.092	99.4
Subregion (RMS)	0.209	---	79.2

The final scene is shown in Figure 2.10. Two surface substructures were detected on two different buildings by the recursive model fitting process; both are roof gables. The gable in the foreground of Figure 2.10 more accurately reflects actual scene structure than does the second gable (which is less accurate due to errors in the DEM).

3. Recovery of Building Geometry from SAR and IFSAR

3.1   Introduction

In recent years, IFSAR-derived digital elevation maps (DEMs) have been used in site reconstruction tasks. SAR interferometry has several advantages over the traditional means of generating DEMs, such as stereo photography or the use of laser altimeters. For instance, optical images can be acquired only during the day and under favorable weather conditions. SAR interferometry, on the other hand, is invariant with respect to the weather, and can be used night or day. The IFSAR sensor can also operate at greater altitudes than most laser scanners.

There are several methods of generating IFSAR data, but we shall consider only the two-antenna, single-pass case here. This means that a single aircraft with two antennae, separated by some known baseline, collects all the data from the scene in a single pass. The phase difference between the two returns (one per antenna) generated by a target on the ground is used to determine that target's 3D position [Leberl 1990]. The Kirtland Air Force Base and MOUT data sets were collected in this manner.

18

IFSAR-derived DEMs are inherently noisy and often have a significant amount of data missing from them. As an example of how inaccuracies arise in the elevation data, consider the effects of layover on the front edge of a building. Layover occurs whenever two or more points are at the same distance from the sensor [Leberl 1990]. In this case, a point P along the front edge of a building will be at the same range as some point G on the ground (see Figure 3.1, left). Because of this, the elevation measured for P will be the average of P's actual height and the height of the ground point G. This phenomenon gives the front edge of a building a "crumbled" appearance (see Figure 3.1, right).

Given the inaccurate and incomplete nature of IFSAR-derived DEMs, much of the previous work done on extracting buildings from other types of DEMs — such as those derived from a stereo pair of optical images — may not be applicable here. This would include, for instance, systems that use parametric models to recognize buildings in the scene [Leberl 1990]. Such systems may not recognize a building after it has been imaged by the IFSAR sensor. That is, the sensor may distort a building's height map in ways the models cannot account for. For instance, the crumbled front edge of a building may make it difficult to fit a stored model to that building. As such, the models employed by the system would need additional parameters to account for the effects of layover. Given that the specific effects of layover are dependent on factors that cannot be anticipated -- such as the material properties of the surrounding terrain -- this may not be a viable solution. Model-based target detection in SAR images, however, has met with some success [Chellappa et al. 1996a,b].

As stated earlier, IFSAR-derived DEMs will have data missing from them. Points for which no return was measured are referred to as "drop-outs." Points are dropped out of the IFSAR image for several different reasons. For instance, a specular target, such as a calm body of water or piece of metal siding, will not have a return measured for it if its surface normal does not point towards the sensor. Points are also dropped out when the slant range image is converted into a grid of elevation values. It is this orthorectification process that creates "layover holes," which can be found near the front edge of a building [Vexcel 1998]. A point can also be dropped out of the image because of an occluding object. For instance, a building's rooftop will occlude the terrain behind it. Because of this, a building will cast a shadow in the image (see Figure 3.2, right). These shadows manifest themselves as large regions of drop-outs in the image, and can be used to detect the presence of buildings in the scene.

In this section we present an algorithm for extracting buildings from an IFSAR DEM. This algorithm operates in two stages. First, the back edges of all of the buildings in the scene are located. These edges are identifiable because of the shadows they cast in the image (see Figure 3.2). Specifically, points that belong to the back edge of a building can be identified by their proximity to a large region of drop-outs. Once the back edge of a building has been found, its rooftop is extracted through region growing. The region's growth begins at this back edge, and proceeds along the building's boundary. A point is added to the growing region only if it belongs to the building's rooftop. This determination is made by comparing the point's height to a threshold found in an elevation histogram of its neighborhood (i.e., through adaptive thresholding). If the

point's elevation exceeds that threshold, it is considered to be part of the rooftop and is added to the growing region. Growth terminates once the region encompasses the building's entire rooftop. This algorithm is suitable for use only on buildings with rectilinear boundaries.

Section 3.2 details the back edge detection process. Section 3.3 describes how a building's rooftop is extracted once its back edge is found. The results of applying this algorithm to the Kirtland AFB and MOUT scenes are shown in Section 3.4.

3.2   Back Edge Detection

3.2.1   Properties of a Back Edge

The back edges of a building are along those walls facing away from the sensor. The rooftop of the building occludes the ground adjacent to a back edge from the sensor, causing no return to be measured for that portion of the surrounding terrain. Therefore, points belonging to a building's back edge can be identified by the shadows they cast in the image (see Figure 3.2). These shadows extend outward from the back edge in the direction of the sensor, where the direction of the sensor is the 2D projection of the axis perpendicular to the flight path (see Figure 3.2, left). Since the occluded area is part of the terrain surrounding the building, we make the assumption that the shadows cast by a back edge will terminate at some point on the ground. This assumption is reasonable in contexts where the buildings are not too closely spaced or surrounded by trees and other obstructions. Thus, a back edgel (which is part of the building's rooftop) will have an elevation greater than that of the point at the terminating end of its shadow (which is part of the ground). This information allows the formulation of two different constraints that any point E must satisfy before being labeled a back edgel:

1. *E* must lie on the border between a rooftop and the shadow it casts.

2. There must be a shadow extending from *E* in the direction of the sensor that terminates at some point *G* belonging to the surrounding terrain. *E* must have an elevation greater than that of the surrounding terrain as represented by the point *G*. The height disparity *dH* between *E* and *G* must be greater than or equal to the minimum height expected of a building (taken here as 3.5 meters).

3.2.2   Characterizing Points on a Building's Back Edge

The process of finding back edgels begins by identifying those points *P* in the image that satisfy the first constraint. Because it is not known a priori which points belong to a building's rooftop, this condition must be approximated. For instance, we could require that such a point border a shadowed region (i.e., a region in the image for which no returns have been measured). All back edgels will have this property because back edges cast shadows in the image.

A more discriminating approximation requires that one must be able to draw a line through P that divides its local neighborhood into an *occluded side* and a *rooftop side*: all points to one side of the line (the occluded side) should be drop-outs, while all points to the other side (the rooftop side) should have a return measured for them. This dividing line represents a segment of the hypothetical back edge to which P belongs. If P is indeed a back edgel, then, in theory, such a line must exist: it runs between the shadow cast by the building (i.e., the occluded side of the dividing line) and the building itself (the rooftop side of the dividing line). Because of the influence of noise, however, the conditions stated above — namely that only dropouts can lie to one side of the edge and only visible points to the other — must be relaxed.

Points that satisfy the above constraint, henceforth known as *shadow edges*, can be identified by applying a series of binary masks to every point in the image for which a return was measured. Each mask M is a disc with a radius of k pixels ($k = 4$ here) and represents the neighborhood of a point on or near a shadow/rooftop border. The dark side of the mask represents the shadow cast by the building's back edge, while the bright side represents the building's rooftop near that back edge. Examples of these masks can found in Figure 3.3.

The *orientation* of a mask points into the occluded, or shadowed, side of the mask. The dividing line (which passes through the mask's center) has an orientation perpendicular to that of the mask's. For example, a binary mask with an orientation of zero has a dividing line that passes through the mask's origin at an angle of 90 degrees. This dividing line represents the hypothetical back edge that passes through the mask's center. All points to the right of that line belong to the occluded side of the mask, while all points to the left of that line belong to the visible, or rooftop, side of the mask (see Figure 3.3). The masks have orientations from 0 to 2π, spaced at 10-degree intervals. This gives us a total of 36 different masks.

Each time a mask is applied to a point P in the image, a disc-shaped window of pixels (with a radius of 4) centered at that point is compared to the mask to generate a match score. One way to compute a match score is to cross-correlate the mask with P's neighborhood. However, this metric is inappropriate given that the mask is binary (i.e., shadow or rooftop). A better approach is to count the number of mismatches S_M^P between the mask M and P's neighborhood. Mismatches occur whenever

- there is a return for a point in the building's shadow (i.e. on the dark side of the mask), as occluded points cannot register a return to the sensor, or

- a point falling into the region reserved for the building's rooftop (i.e., the bright side of the mask) is a drop-out, since presumably the point is not occluded and should therefore have returned the emitted signal.

When determining whether or not a point P is a shadow edge, a set $S^P = (S_0^P, S_1^P, ..., S_{35}^P)$ of 36 such scores are generated, one for each mask in the set of all masks ($M_0, ..., M_{350}$).

Note that the logic expressed in the second condition is somewhat flawed since there are other situations in which a target on the ground will not produce a return (see the Introduction). These masks instead represent the neighborhood of a border point under ideal conditions (i.e., no noise or other distortions).

Figure 3.4a shows an IFSAR image of one of the buildings in the Kirtland scene. When the binary mask M_0 (Figure 3.4b) is applied to this image, those points along the building's right-most edge received the best match scores (Figure 3.4d). This is because the edge bordered a large region of shadowed pixels (i.e., drop-outs) and had an orientation perpendicular to that of the mask's. However, when the mask M_{270} (Figure 3.4c) was applied to the same image, those points along the building's bottommost edge received the best match scores (Figure 3.4e). The two masks generated significantly different responses because of their different orientations: M_0 detects vertical back edges while M_{270} detects horizontal back edges.

3.2.3   Locating Shadow Edges in the Image

The match scores produced by the masks can be used to determine if a point P and its neighborhood (defined earlier as a disc with a radius of four pixels) are consistent with the hypothesis that they belong to a shadow/rooftop border. Specifically, they can be used to determine if P is consistent with the hypothesis that a building back edge E_θ passes through it, where the hypothetical back edge is characterized by a single scalar θ (given in radians). The orientation of the hypothesized back edge is perpendicular to θ, while θ itself points into the edge's shadow.

The determination as to whether or not P is consistent with the hypothesis that a back edge similar to E_θ passes through it is made by comparing the set of match scores S^P observed for P to those one would expect to observe for a point along the hypothesized back edge E_θ. That is, the set of match scores observed for P are compared to the set of match scores S^E one would expect to observe for P under the assumption that E_θ passes through P. If the observed scores are similar to the expected scores, then it is plausible that P belongs to a back edge similar to E_θ.

Computing the Expected Match Scores S^E

Ideally, the back edge E_θ will neatly bisect P's local neighborhood into an occluded side (i.e., shadow) and a visible side (i.e., rooftop). That is, all of the points to one side of this edge will lie in the building's shadow. There will therefore be no returns measured for

these points. Points on the other side of this edge, however, will belong to the building's rooftop and, as such, be in full view of the sensor. These points will therefore have returns measured for them. It is clear, then, that P's neighborhood will be *identical* to the binary mask M_θ [2] if E_θ does indeed exist. As such, the set of match scores S^E expected for a point along a back edge such as E_θ can be computed by comparing each of 36 masks to M_θ. The match score produced by the application of one mask M_ϕ to another mask M_θ is given by the following equation:

$$S_\phi^E = \left(\frac{2 \cdot |\Delta\theta|}{2\pi} \right) \left(\pi r^2 \right) = |\Delta\theta|\, r^2 \tag{1}$$

where r is the radius of the masks (here, r is 4) and $\Delta\theta$ is the difference between the mask's orientation ϕ and θ. This equation will give the set of expected match scores S^E = $(S_0^E, S_1^E, \ldots, S_{35}^E)$ for a point along the hypothetical back edge E_θ.

Note that the set of expected match scores was computed under the assumption that there was no noise or other distortions present in the image (i.e., S^E is the set of scores observed under ideal circumstances). This is quite obviously not the case in a real IFSAR image. In using S^E as the basis of comparison, the issue is whether or not the observed scores are good approximations of the ideal scores. If the approximation is close enough, it is plausible that the observed scores are the ideal scores permuted by noise.

Comparing the Observed Scores to the Expected Scores

The set of match scores S^P derived from the image is compared to the set of ideal match scores S^E S^E using the chi-squared error for binned distributions:

$$x^2 = \sum_{i=1} \frac{\left(N_i - n_i \right)^2}{n_i} \tag{2}$$

where N_i is the number of events observed in bin i and n_i is the number of events expected to be in bin i. In this case, each binary mask M_ϕ has a corresponding bin, and events occur whenever there is a mismatch between M_ϕ and the neighborhood to which it was applied (i.e., the bin count for a mask is equal to its match score — see Section 3.2.2). The observed match scores S_ϕ^P are then compared to the expected match scores S_ϕ^E as follows:

---

2 By this we mean that there are no mismatches between M_θ and P's neighborhood

$$x^2 = \sum_{\phi=0}^{35} \frac{\left(S_\phi^P - S_\phi^E\right)^2}{S_\phi^E} \qquad (3)$$

A chi-squared distribution with 36 degrees of freedom is used to compute the likelihood p that this large of an error could be generated by chance. If p is greater than 0.05, the two sets are considered to match.

Procedure for Finding Shadow Edges

To find back edges at all possible orientations, the set of back-edge hypotheses must have orientations θ that span the range $0°$ to $360°$. This range is broken into ten degree intervals and each interval is represented by a different back-edge hypothesis E_θ. This yields a set H of 36 different back edge hypotheses E_0, E_{10}, E_{20},..., E_{340}, E_{350}. However, because of constraints imposed by the geometry of the sensor, only building edges at certain orientations can be back edges. Thus, all 36 hypotheses do not need t be tested: if a building edge with an orientation of θ could not possibly be a back edge, then the hypothesis E_θ can be removed from H. Specifically, any hypothesis E_θ whose orientation θ faces towards the sensor can be removed from H. This is because building edges with walls facing the sensor do not cast shadows and are therefore not back. For instance, if the sensor direction is vector $[1,0]^T$, then H would have hypotheses $E_0, E_{10},..., E_{80}$ and $E_{270},..., E_{350}$. This constraint cuts the number of back edge hypotheses we must try in half.

The overall procedure for determining whether or not a point is a shadow edge is as follows:

1. Apply the masks to P to generate the set of observed match scores S^P.

2. Select a back-edge hypothesis E_θ from H that has not already been tried. If there are none left, terminate.

3. Compute the chi-squared error between the observed match scores S^P and the match scores S^E expected for a point on our hypothetical back edge E_θ using Equation 4.

4. If the chi-squared error yields a p value greater than 0.05, label P as a shadow edge and terminate. Otherwise, return to step 2.

The chi-squared error of step 3 is computed as follows:

$$x^2 = \sum_{i=0}^{35} \frac{\left(S_i^P - |\Delta\theta|r^2\right)^2}{|\Delta\theta|r^2} \tag{4}$$

where $\Delta\theta$ is the difference between the mask M_i's orientation and θ.

3.2.4   Confirming Shadow Edges and Grouping them into Back Edges

We next determine which shadow edges (Figure 3.5, leftmost) belong to the back edge of a building's boundary as opposed to, say, a layover hole. Such shadow edges can be identified by their compliance with the two constraints given in Section 3.2.1. If a shadow edge E is indeed a back edgel, then, according to the second constraint, E must cast a shadow in the direction of the radar that terminates at some point G on the ground. G is found by moving a small window along a path that begins with E and follows the direction of the sensor. The search terminates when the majority of the pixels within the window have measured returns (i.e., when the window has moved outside of the shadow cast by the building). To overcome the noise inherent in a SAR-derived DEM, the median elevation value of the points in that window is selected as the elevation for G. The elevation value for E is selected in a similar fashion. If the candidate E has an elevation sufficiently greater than that of G, the candidate is selected as belonging to a building's back edge. The difference in elevation between E and G must be greater than (or equal to) the minimum height expected of a building in the scene. Here, we expect the height of a building to be at least 3.5 meters. The shadow edges produced earlier (Figure 3.5, leftmost) will serve as our back-edge candidates. These are then verified using the elevation constraint described above (Figure 3.5, second from the left). Those shadow edges that were not upgraded to back edgels (i.e. those shadow edges that could not satisfy the second constraint given in Section 3.2.1) are stored for later use.

Next, the back edgels are grouped into connected components that represent back edges. This is done in two stages. In the first stage, the system interpolates between verified back edgels. Interpolation occurs along those shadow edges that have met the first criterion but not the second (i.e., those shadow edges that were not promoted to back edgels in the prior step — see above). Such edges are promoted if and only if they form a line with back edgels detected in the previous step (shown in Figure 3.5, second from the left). After interpolation has occurred, a morphological closing is used to bridge small gaps between back edgels. A disc with a radius of two pixels is used as the structuring element in the closing. The resulting back-edge regions are shown in the rightmost panel of Figure 3.5.

Finally, the orientation of the back-edge is ascertained by fitting a line to it. This line is fit using a Hough transform. The accumulator array used in the transform has two axes, R and θ: R is the perpendicular distance from the line to the origin and θ is the angle that perpendicular ray makes with the x-axis. As such, the line corresponding to the accumulator cell (r, ϕ) is given by the following equation:

$$x\cos(\phi) + y\sin(\phi) = r.$$

θ is broken into ten degree intervals, while R is broken into five-pixel intervals. The orientation of a building's back edge is used when fitting a rectangle to that building's rooftop.

3.3   Boundary Detection Through Region Growing

3.3.1   Overall Strategy

After the building's back edge has been detected, the remainder of its boundary is extracted by identifying those points on its rooftop that are near or on one of its bounding edges. This portion of the building's rooftop is located by using a region growing technique that classifies points as belonging to either the ground or the rooftop based on an elevation histogram of their local neighborhood. Points that have been labeled as rooftop are added to the growing region only if they are adjacent to points labeled as ground. In this way, the region's growth is restricted to proceed along the building's boundary (see Figure 3.6). The region growing process is seeded using the back edges extracted earlier.

As mentioned above, classification decisions are based on a threshold found in an elevation histogram of the neighborhood surrounding a point. Since the region's growth is restricted to points near an edge of the building, these neighborhoods will contain points from both the rooftop and the surrounding terrain. Therefore, an elevation histogram of such a neighborhood should have fairly distinguishable modes corresponding to the rooftop and the ground, allowing a suitable threshold to be found between these modes (see Section 3.3.2). Note that it is possible for other structures adjacent to the building to be included in the rooftop region if these structures are also elevated above the local terrain (e.g., trees).

The region growing process is fairly straightforward. A portion of the building's back edge serves as our initial rooftop region. It does not matter where this segment is located on the back edge, so long as it is 8-connected. These points are then labeled as rooftop and added to the list of available seeds, which is initially empty. Next, the points at the terminating ends of the shadows cast by these back edgels are labeled as ground (i.e., the ground points G derived in Sections 3.2.1 and 3.2.4). The rest of the image remains unclassified.

At each iteration, a point is removed from the list of seeds. Unclassified points within an adaptively sized window centered at this seed will be assigned labels (either *rooftop* or *ground*) during this iteration. This window will be made large enough to include several points that have already been labeled as ground. Once the size of the window has been determined, an elevation histogram of the unclassified points within the window is taken and a threshold is selected. This threshold will be used to classify the unlabeled points as either ground or rooftop. The mean elevation of the points within this window that have

already been classified as ground will be used to guide the selection of this threshold (see the next paragraph). Points that have been labeled as rooftop are then added to the list of seeds provided they are adjacent to points classified as ground. This process repeats until no suitable seed points remain.

3.3.2 Threshold Selection Using A Priori Classifications

Since the terrain adjacent to a building is typically flat (at least locally), points belonging to the ground within the same local neighborhood should have similar characteristics. The points within the classification window that have already been labeled as ground can therefore provide a rough estimate of the elevation of any ground point within that window, including those yet to be labeled as ground. As such, the mean elevation μ_{prior} of those points already labeled as ground can aid us in selecting an appropriate threshold t. Specifically, the mean elevation μ_t of those unclassified points that *would* be labeled as ground by a particular threshold t should be approximately the same as the mean elevation μ_{prior} of those points *already* labeled as ground. Note that the two means μ_t and μ_{prior} need not be identical. This will allow for a small gradient in the elevation of the ground plane.

After the window size has been selected, we generate the elevation histogram and compute the mean elevation μ_{prior} of those points within the window that have already been labeled as ground. Next, any local minima t within the elevation histogram are identified and added to the set of all such minima S_{minima}. These local minima will serve as the set of candidate thresholds. Finally, for each t in S_{minima}, we then identify those unclassified points with elevations less than t and compute their mean elevation μ_t. That is, we compute the mean elevation of the points that would be labeled as ground by our candidate threshold t. Once this has been done, our threshold is the elevation t in S_{minima} that minimizes the absolute value $|\mu_{prior} - \mu_i|$. After the threshold has been selected, classification is performed. Note that in the early iterations of the process, the ground points G used to validate the back edgels will provide estimates of the ground's elevation.

An example can be seen in Figure 3.7. There are two local minima in this histogram, indicated in black. The first, or minima A, is at 102.672 meters, and the second, or minima B, is at 104.683 meters. The mean elevation μ_{prior} of the points already classified as ground is 102.199 meters.

The mean ground elevation if minima A was used as our threshold would be 102 meters, only 0.199 meters away from our μ_{prior} of 102.199 m. The mean ground elevation if minima B was used as our threshold would be 102.96 meters, which is 0.761 meters away from our μ_{prior}. Thus, a threshold of 102.762 m is selected.

3.4 Results

Boundaries were established for each building found in the image by placing a bounding box around the rooftop region grown for that building. These bounding rectangles were at an orientation equal to that of the building's back edge. The resultant fits for selected areas of the MOUT and Kirtland scenes are shown in Figures 3.8 and 3.9. Eight of the eleven buildings in the Kirtland scene were detected along with one false positive. Twelve of the fifteen buildings in the MOUT scene were detected. There were no false positives in the MOUT scene.

Buildings A and B in the MOUT site were missed because the shadows they cast extended to the front edge of another building. As such, the shadow cast by any point on either building's back edge will not terminate on the ground. Instead, it will terminate at a point on another building's rooftop. Because the shadows cast by A and B both terminate on the rooftop of a taller building, any point E on either buildings's back edge will have an elevation less than that of the point G found at the terminating end of its shadow. Therefore, none of the points on either buildings' back edge will meet the second criterion required of a back edgel (see Section 3.2.1). As such, neither A nor B's back edge was detected.

Buildings A, B, and C were not detected because the majority of the points corresponding to those buildings were dropped out of the IFSAR image. That is, no returns were measured for most of the points corresponding to buildings A, B, and C. As such, the height of their rooftops could not be determined, making it impossible to detect their back edges. Figure 3.10 shows an optical image of the buildings A, B, and C. Those points that were dropped out of the IFSAR image are indicated in black. It is evident from this figure that buildings A, B, and C were simply not detected by the IFSAR sensor. The only false positive of both data sets occurred at site D of the Kirtland scene, and is shown in Figure 3.9. From the optical image of that scene, it appears that D may be some sort of construction site. If this is the case, then it is possible that a foundation erected at that site cast a shadow in the IFSAR image. This would lead to the detection of a back edge at site D. This back edge would then serve as the seed of the false positive.

The building hypotheses generated for the MOUT site (Figure 3.11, left) were compared to the set of reference polygons shown in Figure 3.11. These polygons were extracted by hand and represent the true boundaries of the buildings in the scene. A rooftop hypothesis extracted from the IFSAR image is valid only to the extent that it overlaps one of these polygons. Two metrics were used to evaluate the boundaries extracted by the system:

$$Detection \ Rate = \frac{TP}{TP + FN}$$

$$False \ Alarm \ Rate = \frac{FP}{TP + FP}$$

where *TP* is the total number of true positives, *FP* is the total number of false positives, and *FN* is the total number of false negatives.

A point inside the bounding box established for a building (Figure 3.11, left) is considered to be a true positive if it is also within that building's reference polygon (Figure 3.11, right). Otherwise, that point is labeled as a false positive. A point is a false negative if it is interior to the reference polygon but outside of the boundary extracted by the system. Table 3.1 shows the false alarm and detection rates for the buildings found by the system. Those buildings that went undetected by the system were not evaluated in this fashion.

Table 3.1 Detection and False Alarm rates for the MOUT site.

Buildings	TP	FP	FN	Detection Rate	False Alarm Rate
0	2229	218	344	0.86	0.09
1	1130	348	48	0.95	0.23
2	1187	877	455	0.72	0.42
3	374	44	214	0.63	0.1
4	3036	1086	84	0.97	0.32
5	1145	115	698	0.62	0.09
6	708	0	426	0.62	0
7	480	115	174	0.74	0.19
8	1191	459	0	1.0	0.28
9	467	0	341	0.58	0
10	295	0	139	0.68	0
11	1310	226	458	0.74	0.15
Mean				0.73	0.16

4. Bibliography

Andersen, S., Olesen, K., Jensen, F.V., and Jensen F. "Hugin — A Shell for Building Bayesian Belief Universes for Expert Systems." *Proceedings of the 11th International Congress on Uncertain Artificial Intelligence*, pp. 1080–1085, 1989.

Aurenhammer, F. "Voronoi Diagrams — A Survey of a Fundamental Geometric Data Structure." *ACM Computing Surveys*, Vol. 23. pp. 345–405, 1991.

Besl, P., and Jain, R. "Segmentation Through Variable Order Surface Fitting." *IEEE T-PAMI*, Vol. 10, pp. 167–192, 1988.

Besl, P., and Jain, R. "Three-Dimensional Object Recognition." *Computing Surveys*, No. 1, pp. 75-145, 1985.

Breese, J. S., and Heckerman D. "Decision-Theoretic Case-Based Reasoning." Microsoft Research, MSR-TR-95-03, January 1995.

Brooks, R. Symbolic Reasoning Among 3-D Models and 2-D Images. *Artificial Intelligence* Vol. 17, pp. 285–348, 1981.

Brown, C., Marengoni, M., and Kardaras, G. Bayes Nets for Selective Perception and Data Fusion. *Proceedings of the SPIE on Image and Information Systems: Applications and Opportunities*, 1994.

Chellappa, A. R., Kuttikkad, S., and Novak, L.M. Building 2-D Wide Area Site Models from Single- and Multi-Pass Polarization SAR Data. *SPIE Proceedings: Algorithms for Synthetic Aperture Radar Imagery III*, p. 34 – 44, April 1996a.

Chellappa, A. R., Kuttikkad, S., Meth, R., Burlina, P., Ome, K., and Shekkar, C. Model Supported Exploitation of SAR Imagery. *Proc. ARPA Image Understanding Workshop*, p. 389–408, 1996b.

Cheng, J., Bell, D., and Liu, W. "Learning Belief Networks from Data: An Information Theory Based Approach." In Proceedings of the 6th ACM International Conference on Information and Knowledge Management, 1997.

Cheng, J., Bell, D., and Liu, W. "Learning Bayesian Networks from Data: An Efficient Approach Based on Information Theory." Tech Report — Department of Computer Science, University of Alberta, 1998.

Cohen, I., Cohen, L., and Ayache, N. "Introducing Deformable Surfaces to Segment 3D Images and Infer Differential Structure." Technical Report, INRIA, 1991.

Collins, R., Cheng, Y., Jaynes, C., Stolle, F., Wang, X., Hanson, A., and Riseman, E. Site Model Acquisition and Extension from Aerial Images. *Proceedings of the International Conference on Computer Vision*, pp. 888–893, 1995.

Collins, R., Jaynes, C., Cheng, Y., Stolle, F., Hanson, A., and Riseman, E. The Ascender System: Automated Site Modelling from Multiple Aerial Images. *Computer Vision and Image Understanding: Special Issue on Building Detection and Reconstruction from Aerial Images*, 1998.

Crevier, D., and Lepage, R. "Knowledge-Based Image Understanding Systems: A Survey." *Computer Vision and Image Understanding*, 67(2), 1997, pp. 161–185.

do Carmo, M. P. Differential Geometry of Curves and Surfaces, Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

Draper, B., Hanson, A., and Riseman, E. "Knowledge-Directed Vision: Control, Learning, and Integration." *Special Issue of Proc. of the IEEE*, No. 11, pp. 1625–1637, November 1996.

Fua, P. "Reconstructing Complex Surfaces from Multiple Stereo Views." *5th International Conference on Computer Vision*, Cambridge, MA, 1995.

Fua, P., and Leclerc, Y. G. "Using 3-Dimensional Meshes to Combine Image-Based and Geometry-Based Constraints." *European Conference on Computer Vision*, pages 281–291, Stockholm, Sweden, May 1994.

Hanson, A., and Riseman, E. "Visions: A Computer System for Interpreting Scenes." In Computer Vision Systems, A. Hanson and E. Riseman, Eds. Academic Press, 1978.

Haralick, R., and Shapiro, L. Computer and Robot Vision. Addison-Wesley, 1993.

Herbert, M., Ikeuchi, K., and. Delingette, H. A Spherical Representation for Recognition of Free-Form Surfaces. *Proc. IEEE T-PAMI*, Vol. 17, No. 7, pp. 681–689, 1995.

Herman, M., and Kanade, T. Incremental Reconstruction of 3-D Scenes from Multiple Complex Images. *Artificial Intelligence*, Vol. 30, pp. 289–341, 1986.

Horn, B.K.P. Robot Vision, MIT Press, McGraw-Hill, Cambridge, Massachusetts, 1986.

Howard, R.A., and Matheson, J. E. "Influence Diagrams," in Readings on the Principles and Applications of Decision Analysis, R. A. Howard, and J.E. Matheson, eds., Strategic Decision Group, Menlo Park, California, 1984, pp. 721–762.

Jaynes, C., Stolle, F., and Collins, R. Task Driven Perceptual Organization for Extraction of Rooftop Polygons. *IEEE Workshop on Applications of Computer Vision* (1994), 152–159.

Jaynes, C., Stolle, F., Schultz, H., Collins, R., Hanson, A., Riseman, E. Three-Dimensional Grouping and Information Fusion for Site Modeling from Aerial Images. *DARPA Image Understanding Workshop*, pp. 479–490, 1996.

Jaynes, C., Hanson, A., Riseman, E., Schultz, H. "Automatic Building Reconstruction from Optical and Range Images." *International Conference on Computer Vision and Pattern Recognition (CVPR'97)*, San Juan, Puerto Rico, 1997a.

Jaynes, C., Collins, R., Cheng, Y.Q., Wang, X.G., Stolle, F., Schultz, H., Hanson, A., Riseman, E. "Automatic Construction of Three-Dimensional Models of Buildings." RADIUS: Image Understanding for Imagery Intelligence, Oscar Firschein and Thomas Strat (Eds.), Morgan Kaufmann Publishers, San Francisco, CA, (Ed.), 1997b, Chapter 3, pp. 223–236.

Jaynes, C., Marengoni, M., Hanson, A., Riseman, E. "3D Model Acquisition Using a Bayesian Controller." *International Symposium on Engineering of Intelligent Systems/EIS'98*, Univ. of La Laguna, Tenerife, Spain, Feb. 11–13, 1998a.

Jaynes, C., Marengoni, M., Hanson, A., Riseman, E. "Intelligent Control for Automatic Model Acquisition from Aerial Images." *IASTED International Conference on Intelligent Systems and Control,* Halifax, Canada, June, 1998b, pp. 30–35.

Jaynes, C., Hanson, A., and Riseman, E. "Recursive Recovery of Three-Dimensional Scenes." *Computer Vision and Pattern Recognition*, Santa Barbara, CA, June 23–25, 1998c.

Jensen, F. An Introduction to Bayesian Networks. Springer Verlag New York, 1996.

Kass, M., Witkin, A., and Terzopoulos, D. "Snakes: Active Shape Models." *Int. Journal of Computer Vision*, Vol. 1, pp. 321–331, 1988.

Krebs, B., Burkhardt, M., and Horn, B. "A Task Driven 3D Object Recognition System using Bayesian Networks." *Proceedings of the International Conference on Computer Vision, Bombay,* India, 1998,

Kumar, V., and Desai, U. "Image interpretation using bayesian networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1), 1996, pp. 74–77.

Leberl, F. W. Radargrammetric Image Processing, Artech House, 1990.

Leonardis, A., Gupta, A., and Bajcsy, R. "Segmentation of Range Images as the Search for Geometric Parametric Models." *Int. Journal of Computer Vision*, Vol. 14, pp. 253–277, 1995.

Lindley, D. V. Making Decisions: Second Edition, John Wiley and Sons, 1985.

Lowe, D. "Fitting Parameterized Three-Dimensional Models to Images." *IEEE T-PAMI*, Vol. 13, pp. 441–450, 1991.

Mann, W. B., and Binford, T. O. "An Example of 3-DIinterpretation of Images Using Bayesian Networks." *Proc. DARPA Image Understanding Workshop*, 1992, pp. 793–801.

Marengoni, M., Jaynes, C., Hanson, A., and Riseman, E. "A Control System for 3D Reconstruction Using Bayes Nets." *Computer Vision and Pattern Recognition*, Santa Barbara, CA, June 23–25, 1998.

Marengoni, M., Jaynes, C., Hanson, A., Riseman, E. "Ascender II, a Visual Framework for 3D Reconstruction." *First International Conference on Computer Vision Systems*, Las

Palmas, Gran Canaria, Spain, January 13-15, 1999, Lecture Notes in Computer Science 1542, Christensen, H. I. Editor, Computer Vision Systems, Springer Verlag, 1999, pp. 469–488.

McKeown, D.M., Harvey, W.A., and McDermott, J. Rule-based interpretation of aerial imagery. *IEEE T-PAMI*, Vol. 7, pp. 570–585, 1985.

Miller, J. and Stewart, C. "Prediction Intervals for Surface Growing Range Segmentation." *International Conference on Computer Vision and Pattern Recognition (CVPR'97)*, San Juan, Puerto Rico, 1997.

Montagnat, J. and Delingette, H. "A Hybrid Framework for Surface Registration and Deformable Models." *International Conference on Computer Vision and Pattern Recognition (CVPR'97)*, San Juan, Puerto Rico, 1997.

Nelder, J., and Mead, R. *Computational Journal*, vol. 7, pp. 308–313, 1965.

Pearl, J. Probabilistic Reasoning in Intelligent System: Networks of Plausible Inference. Morgan Kaufmann, 1988.

Rimey, R., and Brown, C. Task-Oriented Vision with Multiple Bayes Nets. In Active Vision, B. A. and Y. Aliomonos, Eds. The MIT Press, 1992.

Strat, T. Employing Contextual Information in Computer Vision. *Proceedings of ARPA Image Understanding Workshop*, 1993.

Taubin, G. "Estimation of Planar Curves, Surfaces, and Non-Planar Space Curves Defined by Implicit Polynomials with Applications to Edge and Range Image Segmentation." *IEEE T-PAMI*, Vol. 13, No. 11, pp. 1115–1138, Nov. 1991.

Terzopoulos, D. and Metaxas, D. "Dynamic 3D Models with local and global deformations: Deformable Superquadrics." *Int. Conference on Computer Vision (ICCV'90)*, pp. 600–615, Osaka, 1990.

Vexcel Corporation. "Building Extraction from IFSAR Data." *Private Communication*, 1998.

Wang, C., and Srihari, S. A Framework for Object Recognition in a Visually Complex Environment and its Application to Locating Address Blocks on Mail Pieces. *International Journal of Computer Vision*, pp. 125–151, 1988.

Weidner, U., and Forstner, W. Towards Automatic Building Extraction from High Resolution Digital Elevation Models. *ISPRS Journal*, p. 38 – 49, 1995.

Weiss, A. M. Analysis of Some Modified CFAR Processors. *IEEE Transactions on Aerospace and Electronic Systems*, p. 102 – 114, Jan. 1982.

Zhang, Z. "Iterative point matching for registration of free-form curves and surfaces." *Int. Journal of Computer Vision*, Vol. 13, No. 2, pp. 119–152, Dec. 1994.

Zhang, D., and Herbert, M. "Multi-Scale Classification of 3-D Objects." *International Conference on Computer Vision and Pattern Recognition (CVPR'97),* San Juan, Puerto Rico, 1997.
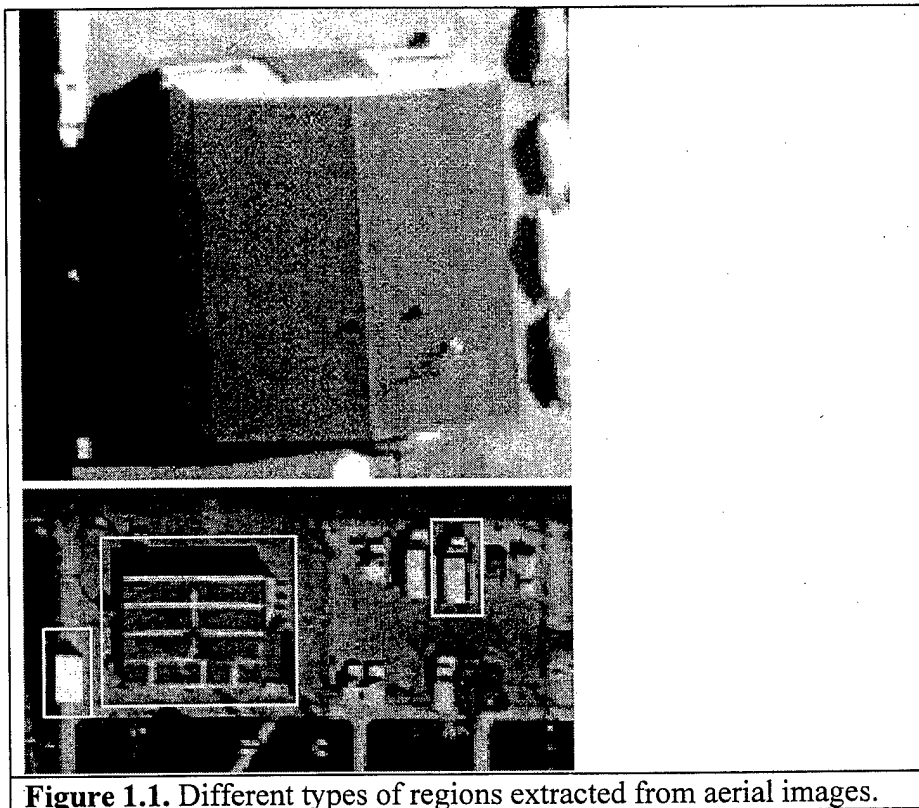
5. Figures

Figure 1.1. Different types of regions extracted from aerial images.

Figure 1.2. Process overview. Decisions are based on current knowledge about the site. Vision algorithms, stored in the visual subsystem, gather evidence about the site, update the knowledge base, and produce geometric models.
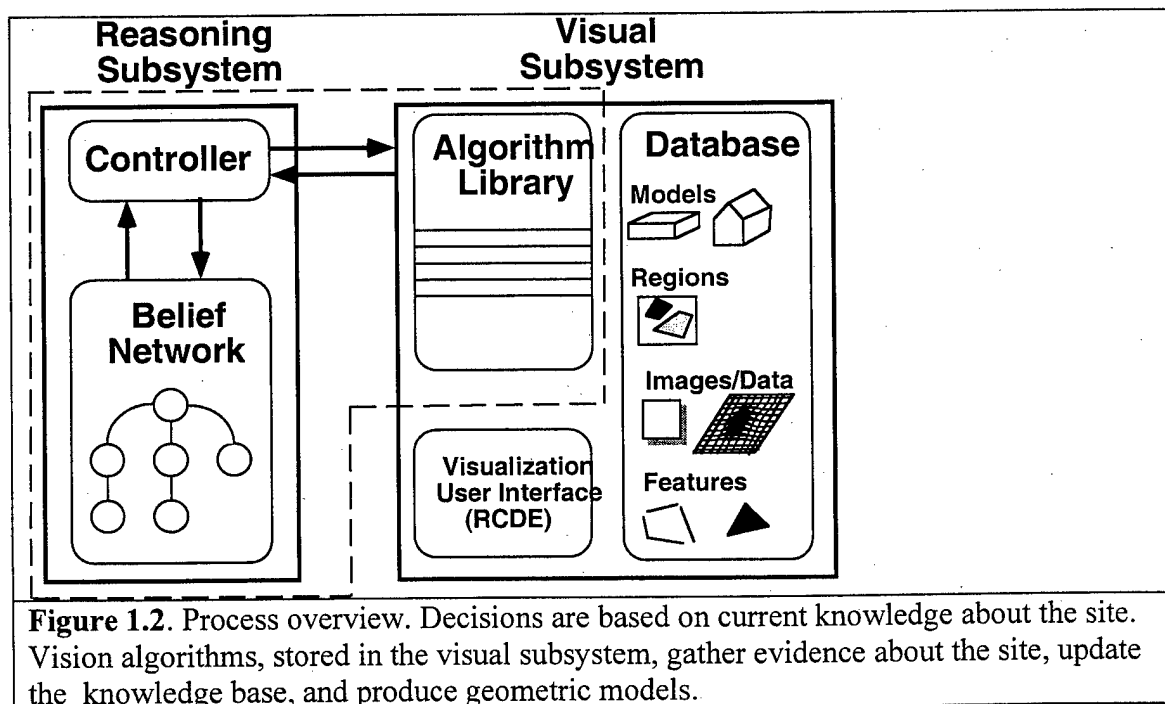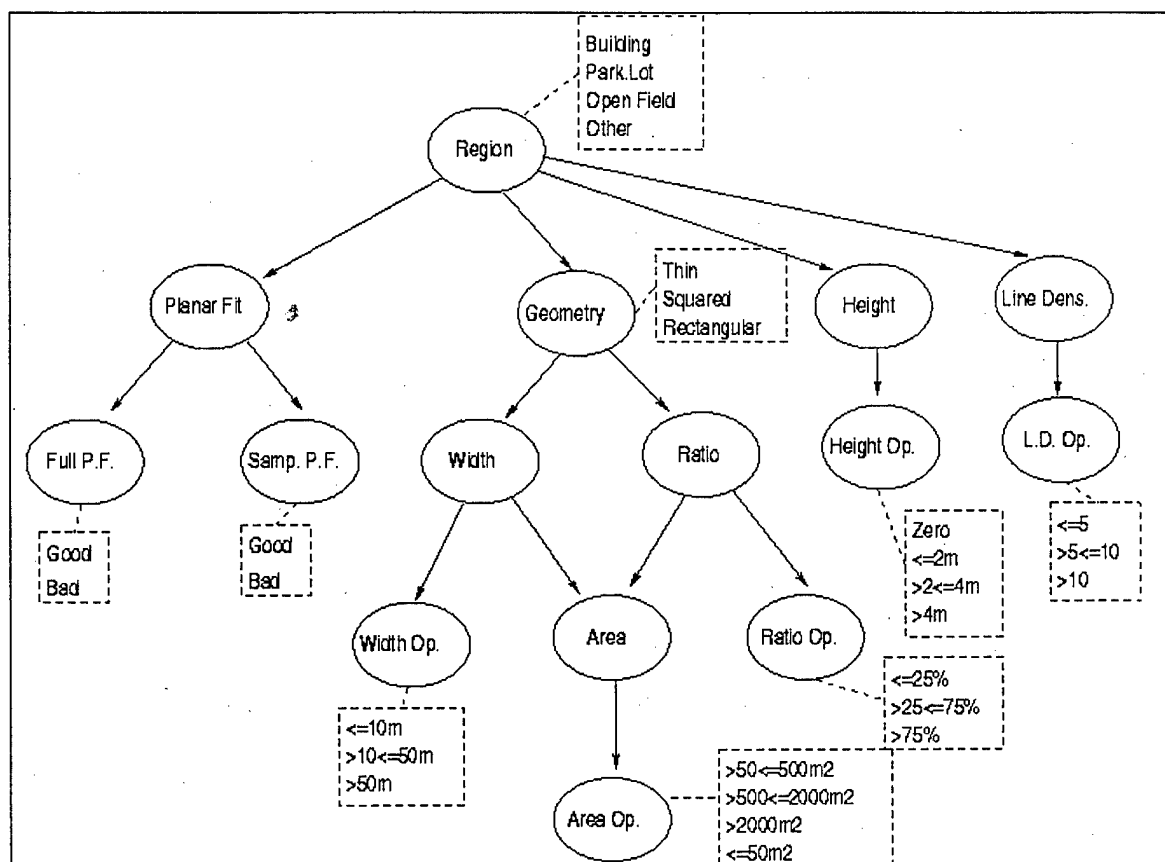
Figure 1.3. The level 0 handcrafted network determines if a region belongs to one of the possible object classes (Building, Parking Lot, Open Field, or Other).
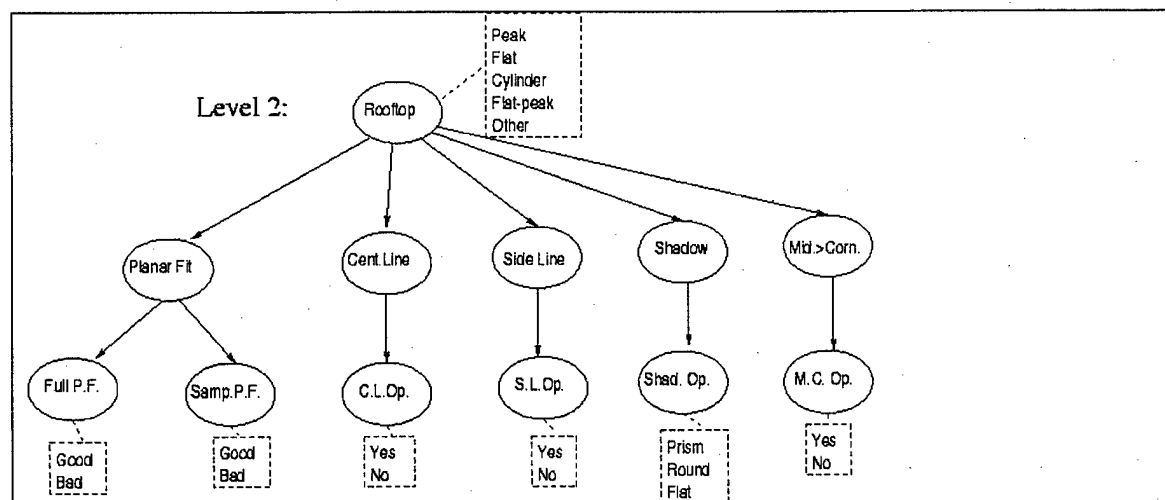
Figure 1.4. The level 2 handcrafted network used to determine the type of rooftop (Peaked, Flat, Flat-Peak, Cylinder, or Other), once a single building is detected.

36

Figure 1.5. A generic Bayesian network for the Ascender II system.

Figure 1.6. The level 0 learned network determines if a region belongs to one of the possible object classes: Building, Parking Lot, Open Field, or Other.
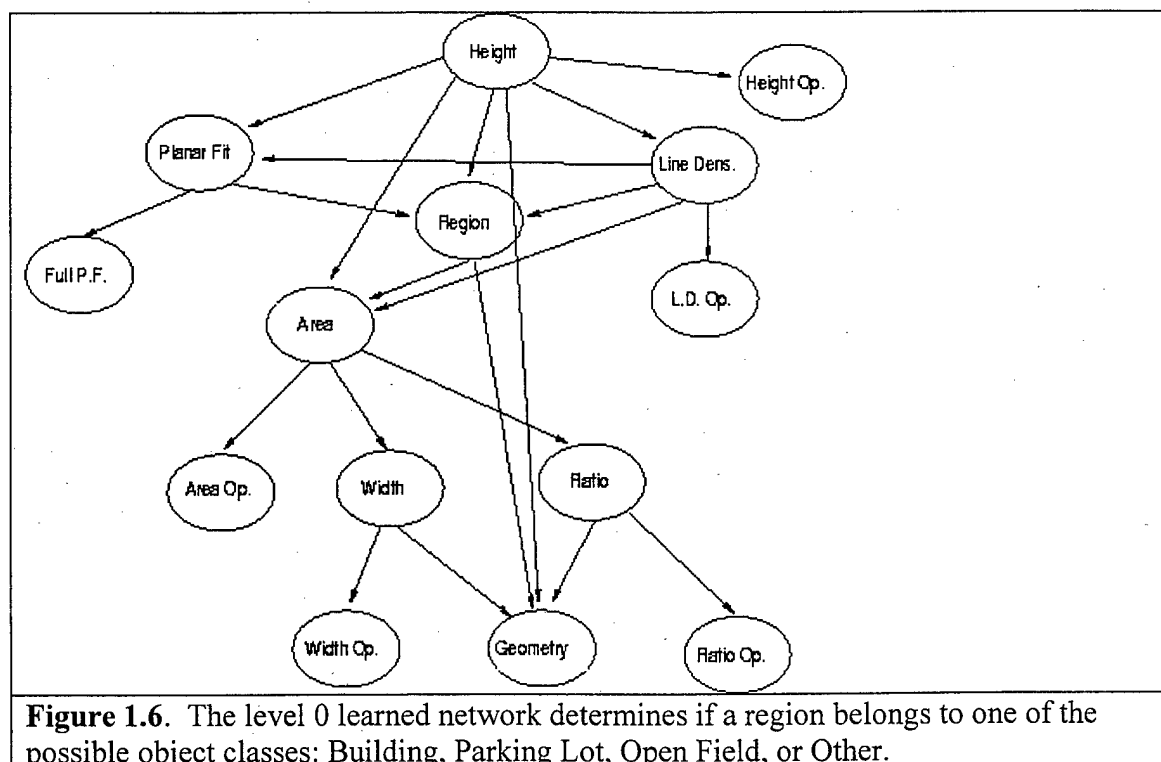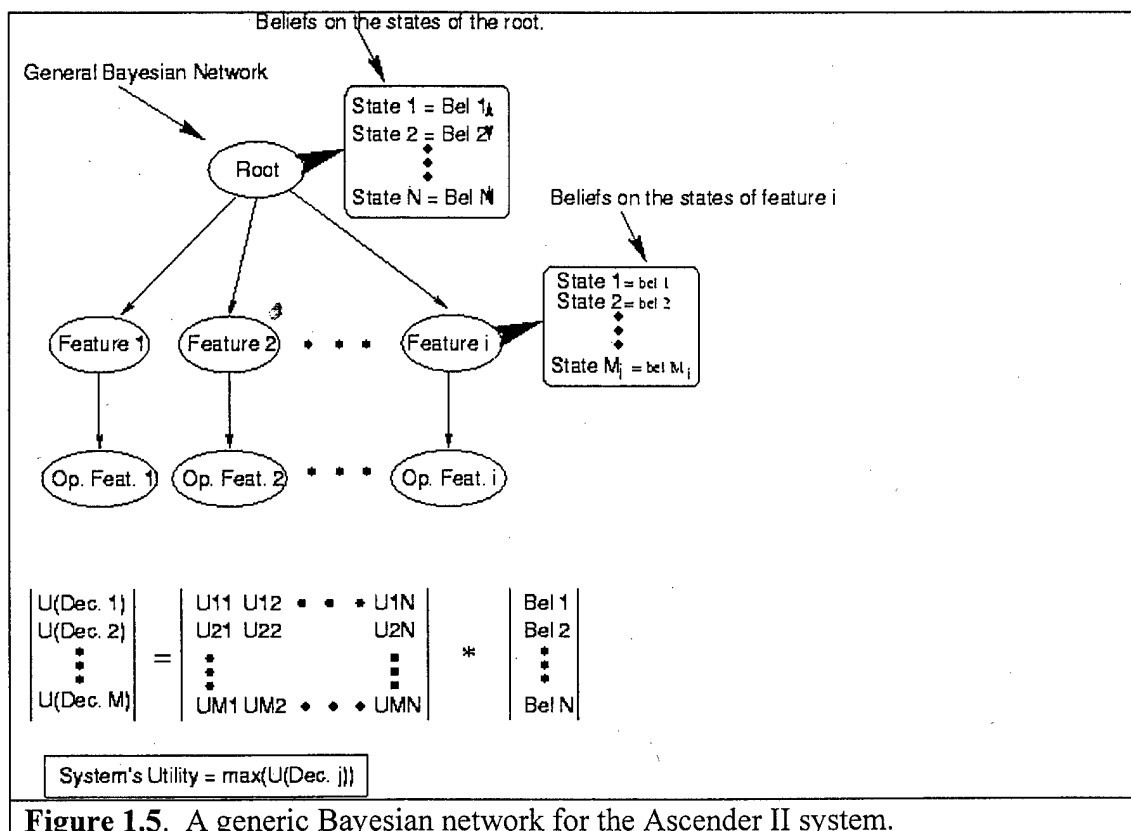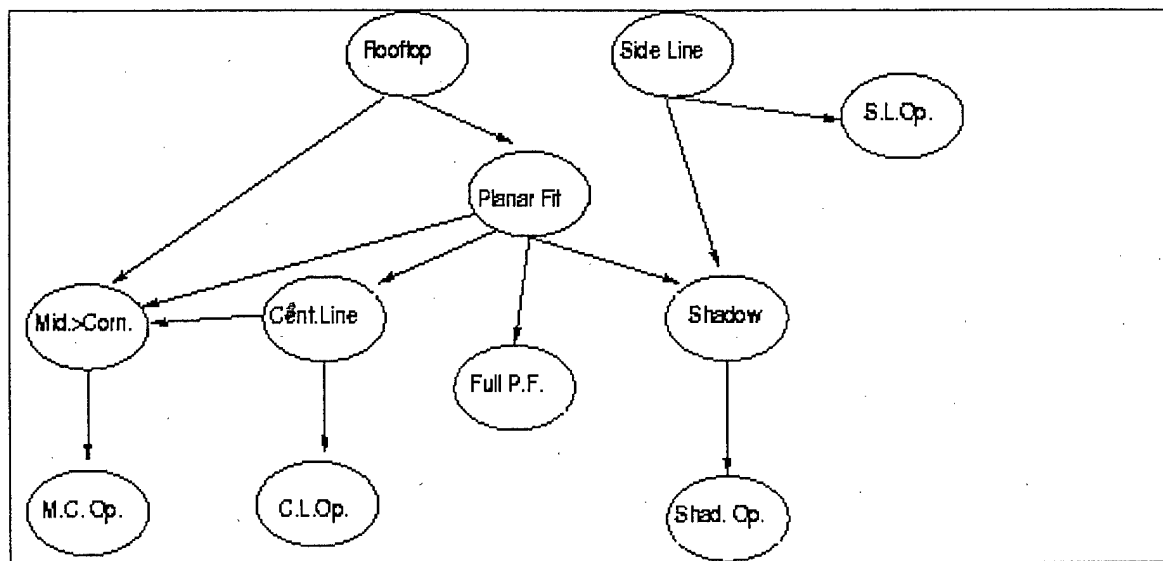
Figure 1.7. This level 2 learned network is called after a single building is detected. It is used to determine the building's rooftop type (Peak or Flat).
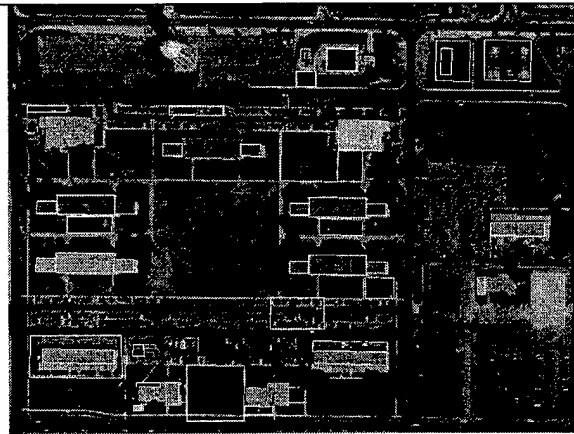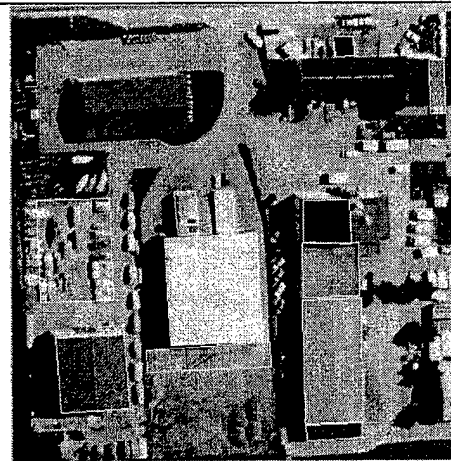
Figure 1.8. Input building hypotheses from the four data sets: (a) Fort Hood, (b) Avenches, (c) Fort Benning, and (d) ISPRS Flat scene. The building hypotheses for the Fort Benning data were created by fusing hypotheses from Ascender I and SAR data (the latter generated by Vexcel Corporation). The Ascender data used in (a–c) were generated by running the original system constrained to detect two-dimensional building footprints. The hypotheses for the flat scene (d) were generated by hand.
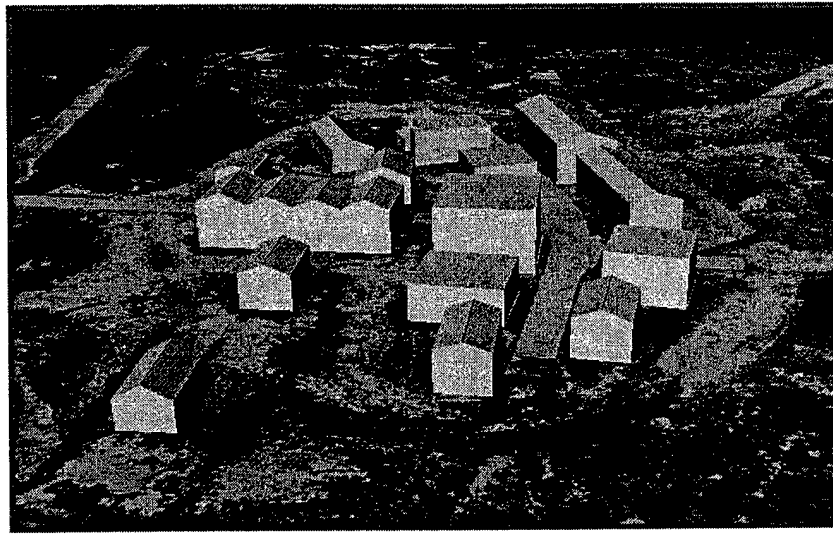
Figure 1.9. Automatic 3D reconstruction from the Fort Benning data set.
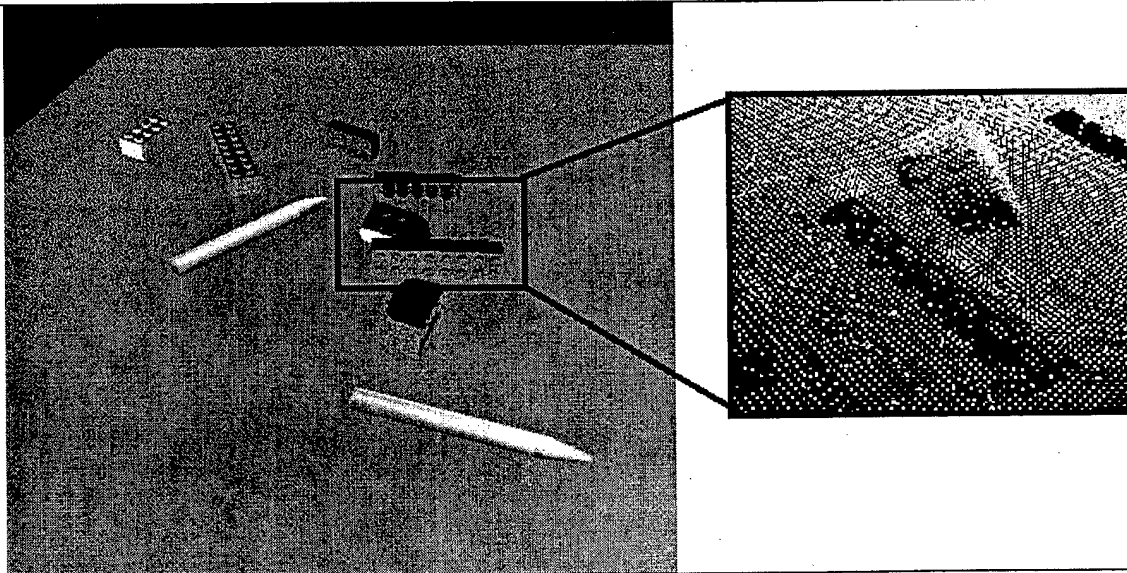
Figure 2.1. Left: Test scene containing nine different objects at various orientations. Right: Close-up of the cloud of 3D points produced from a synthetic range sensor model and corrupted with Gaussian and random noise.
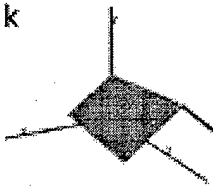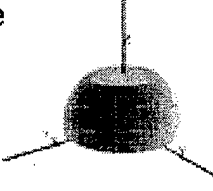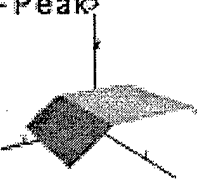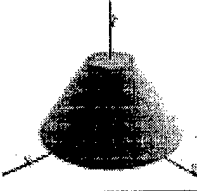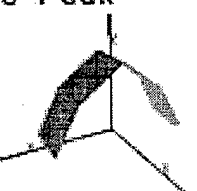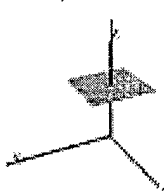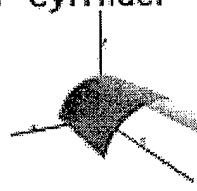
Figure 2.2. Surface model class library. The number of free parameters for each model class is shown at left. For example, the peak model contains two free parameters, its distance along some vector and the angle between the two component planes.
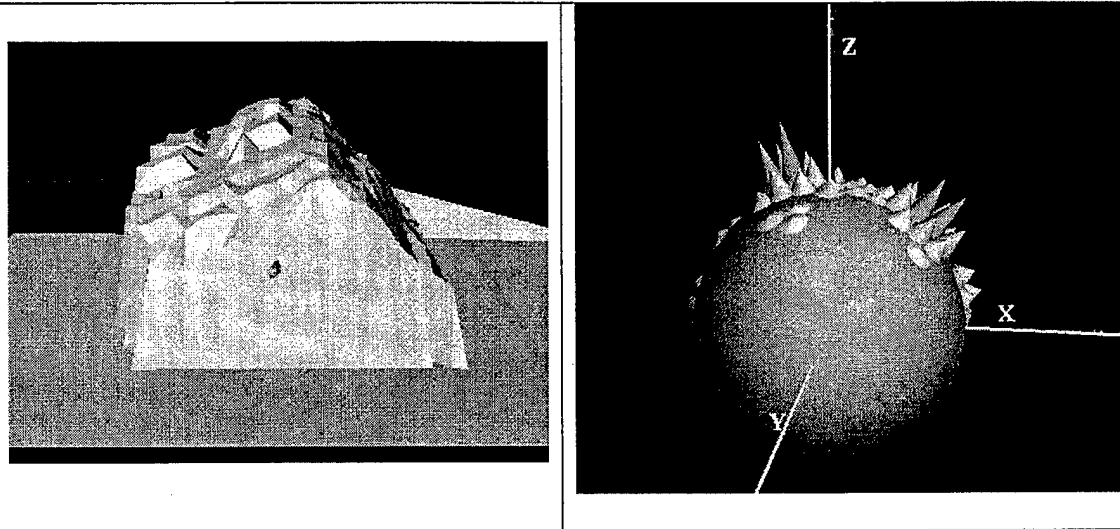
Figure 2.3. (a) Surface mesh fit to region 4 of the toy scene. The object is a tilted 4-peg Lego block. (b) Constructed Gaussian image.

Figure 2.4. Maximum response of correlation score about the 49 different axes of rotation. The rotation vector (0.383,0.0,0.924), shown as a darkened column (top), correlated maximally with the data with a response of 0.82 (shown in the bottom graph).
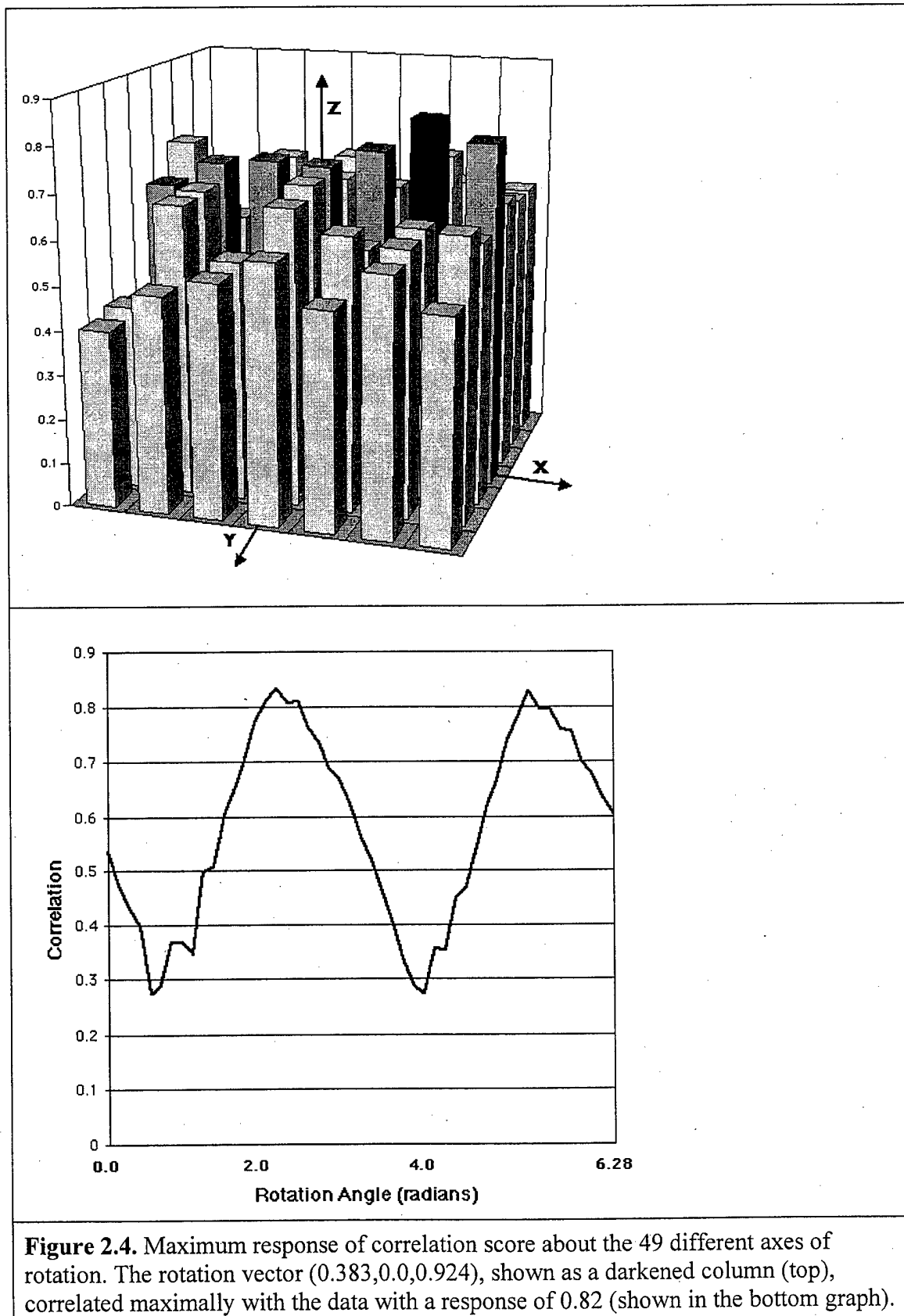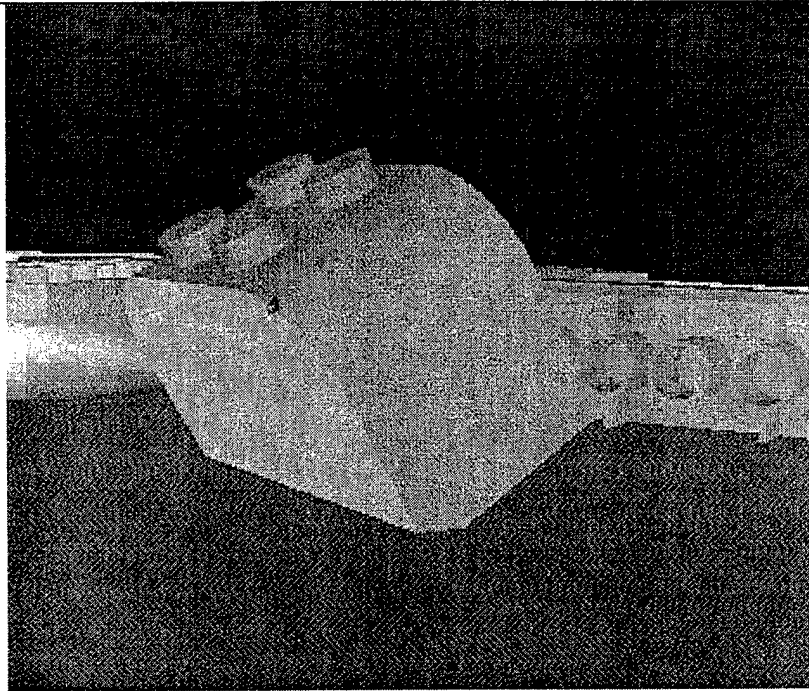
(a)
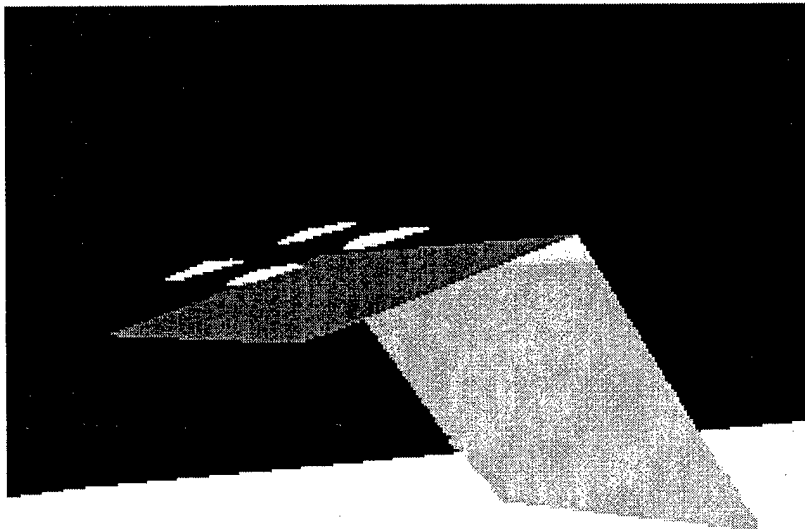
(b)

Figure 2.5. (a) Close-up view of object #4. Note: The object occluding object #4 has been removed to allow a clear view for comparison. (b) Reconstructed surface of region 4. Note that subregions have also been detected and reconstructed (see outlier clustering).
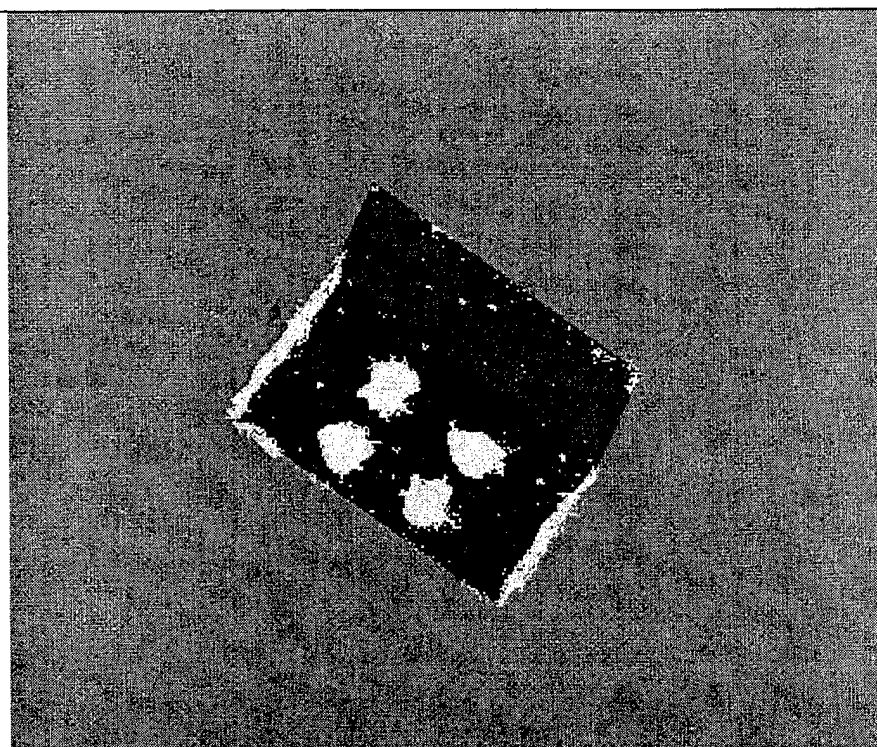
(a)
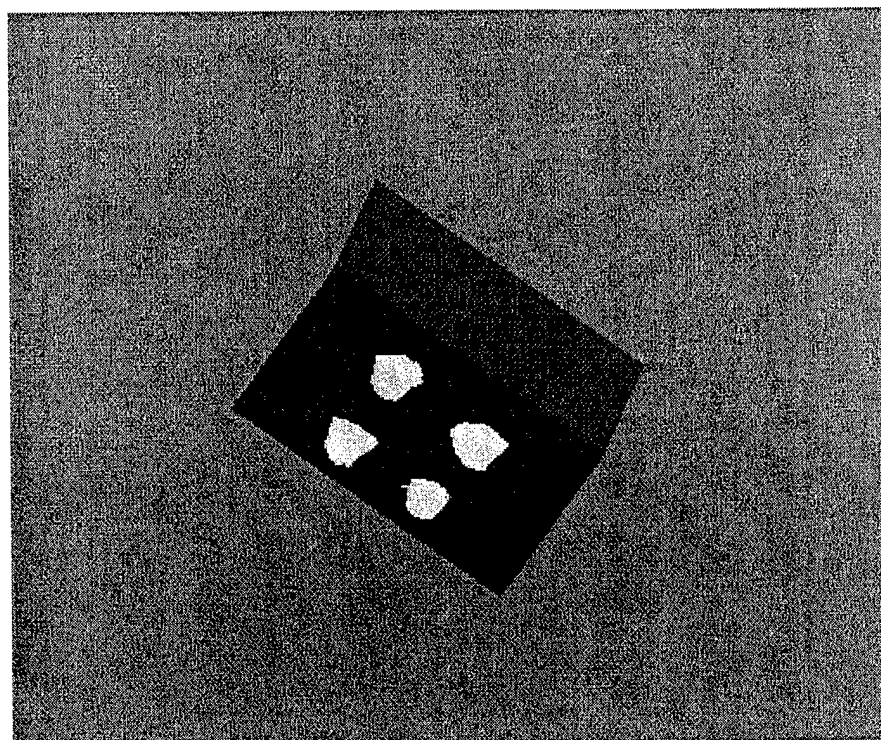
(b)

Figure 2.6. (a) Outliers with respect to the model fit within region 4. (b) Remaining outlier regions after clustering.
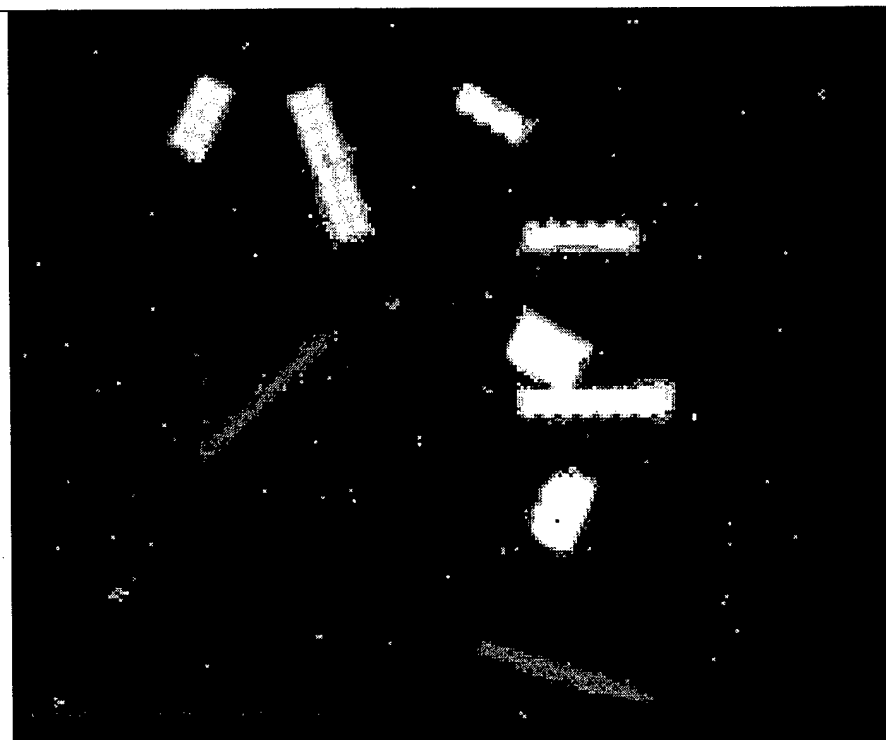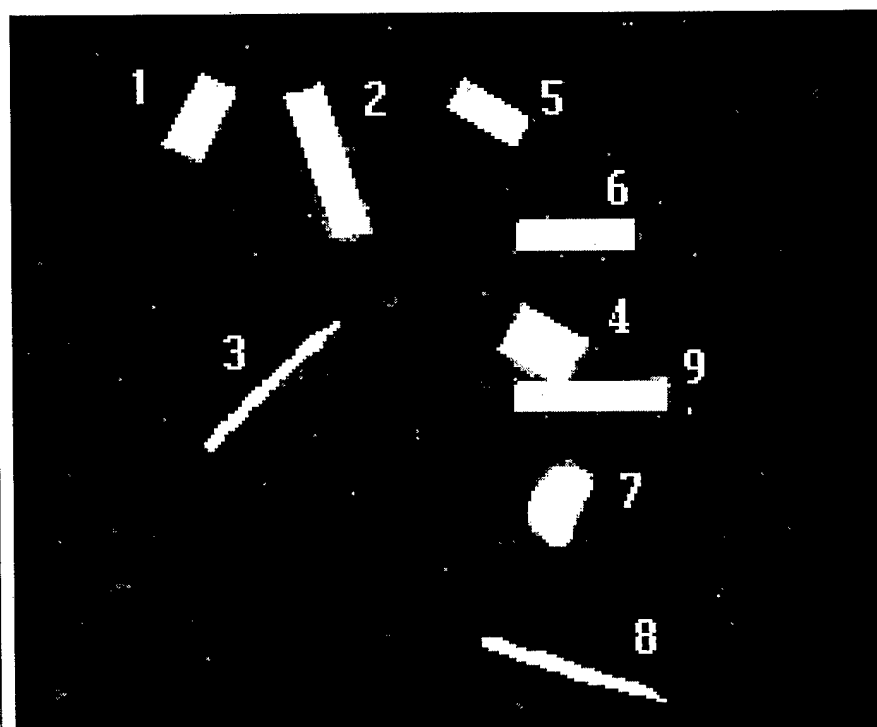
Figure 2.7. (a) Range image used for scene reconstruction. (b) Nine detected regions after region 0 (ground plane) has been fit.

46
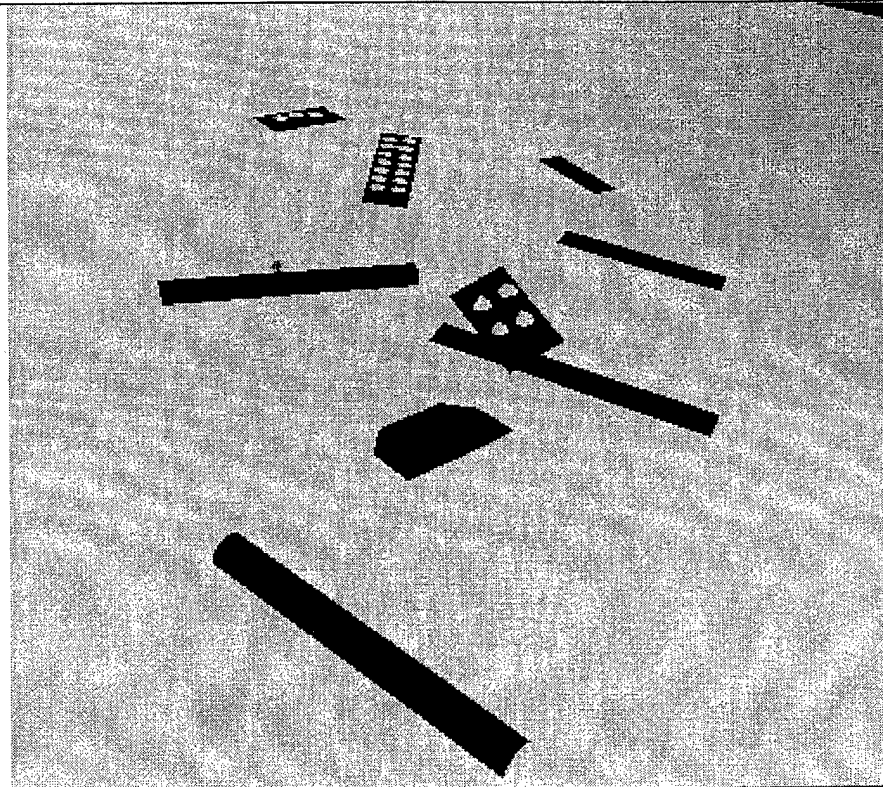
Figure 2.8. Reconstructed surfaces of the "tabletop" scene.

(a)

(b)

Figure 2.9. (a) Image of the Ascona region used for reconstruction. (b) Corresponding DEM recovered from stereo processing.
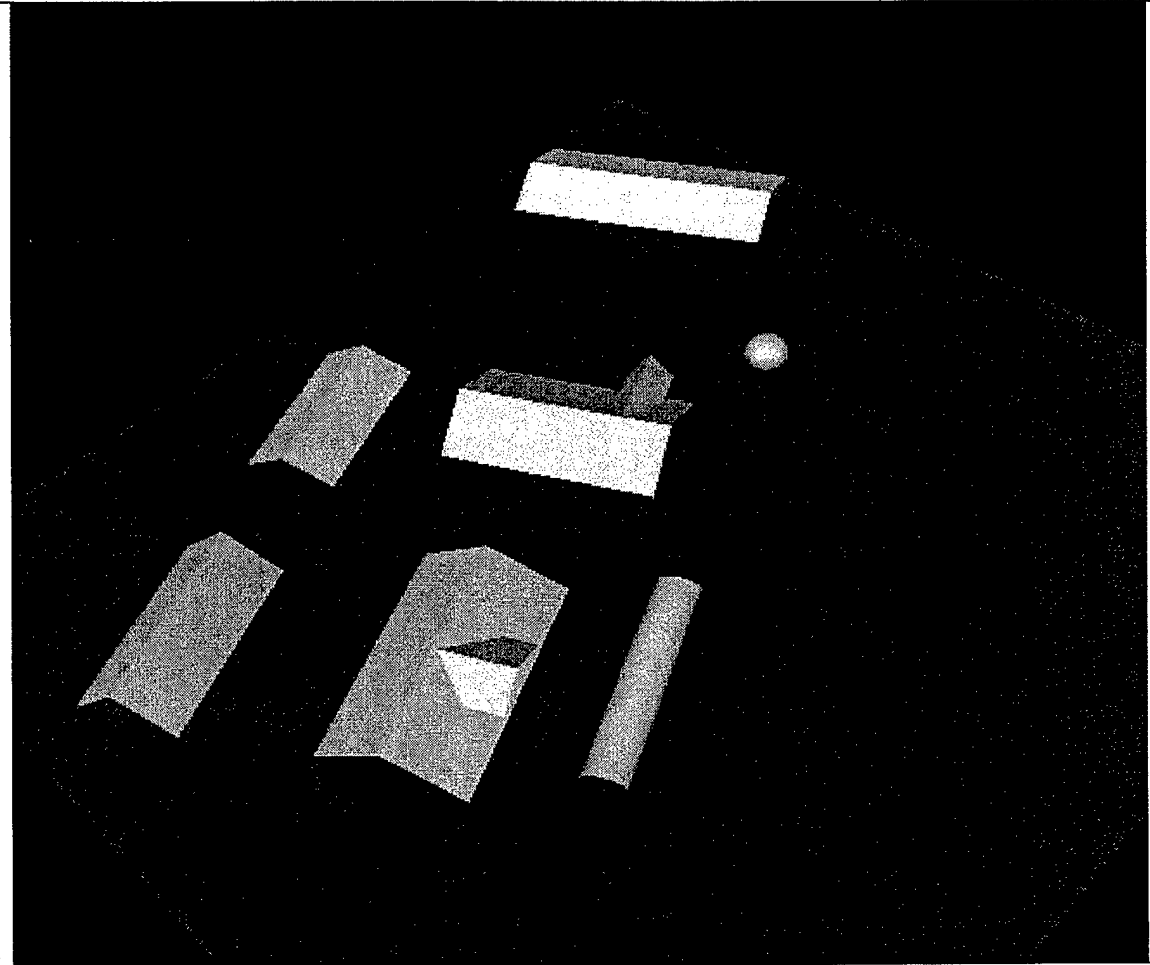
Figure 2.10. Reconstructed scene. All buildings and two rooftop substructures were recovered. Two areas of treetops converged close enough to a cylinder and dome model to be reconstructed.
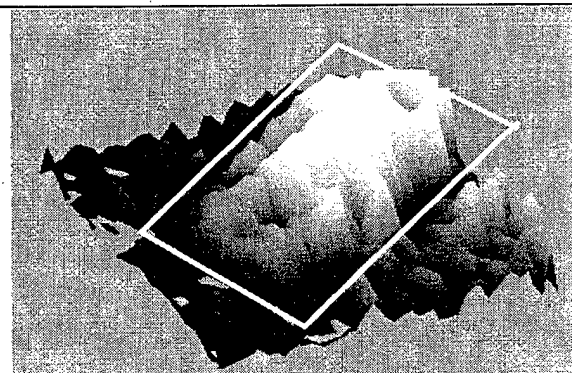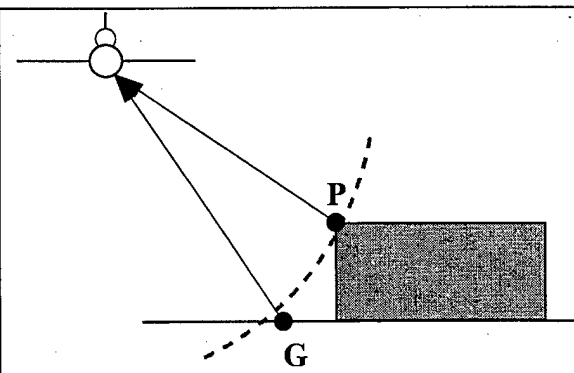

Figure 3.1: Left: Point G on the ground is at the same range as point P on the rooftop. Right: Height map of a building. The building's boundary is shown in white. The darker values at the building's front edge indicate that it is at a lower elevation than the rest of the rooftop.
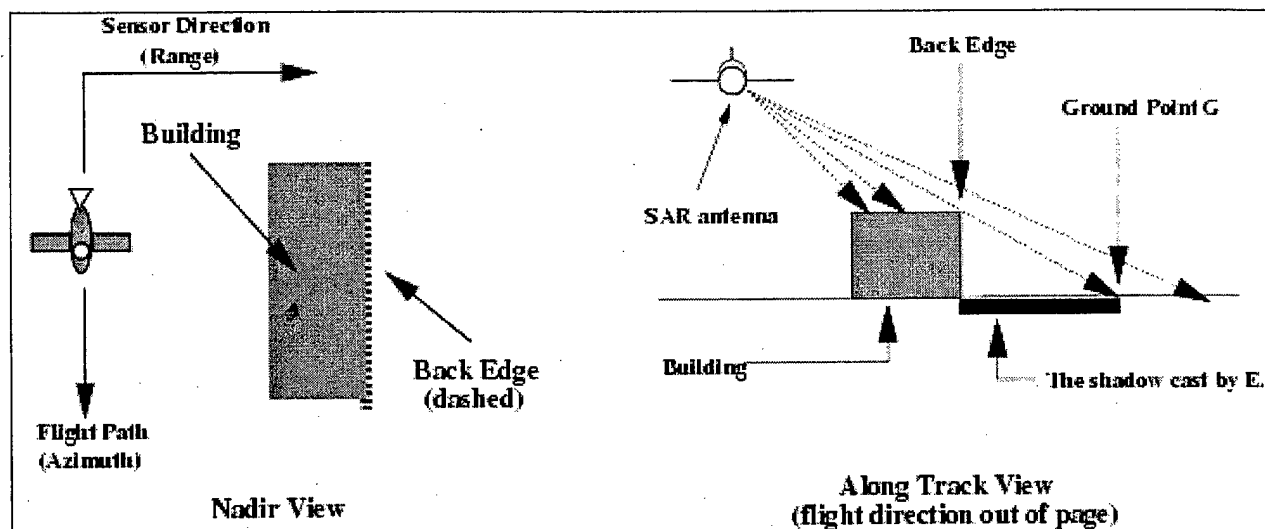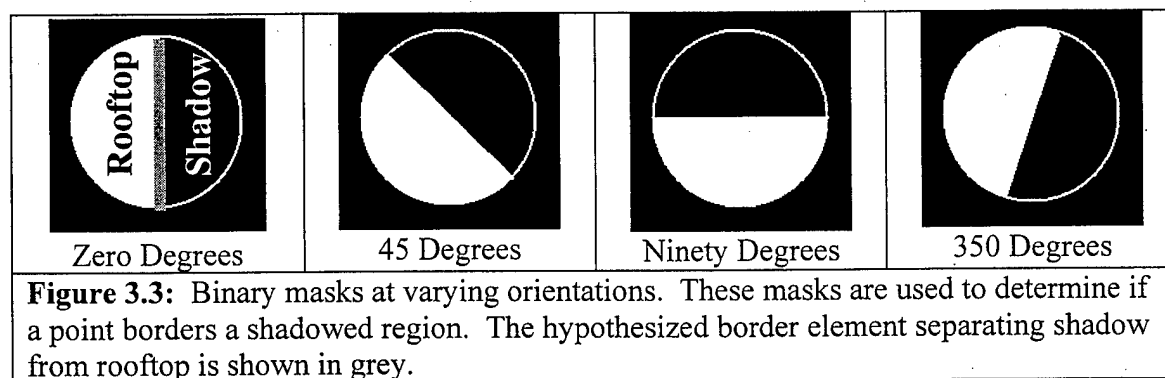
Figure 3.2: Geometry of SAR data acquisition. The shadow cast by a building's back edge extends from back edgel E to a point G belonging to the surrounding terrain.
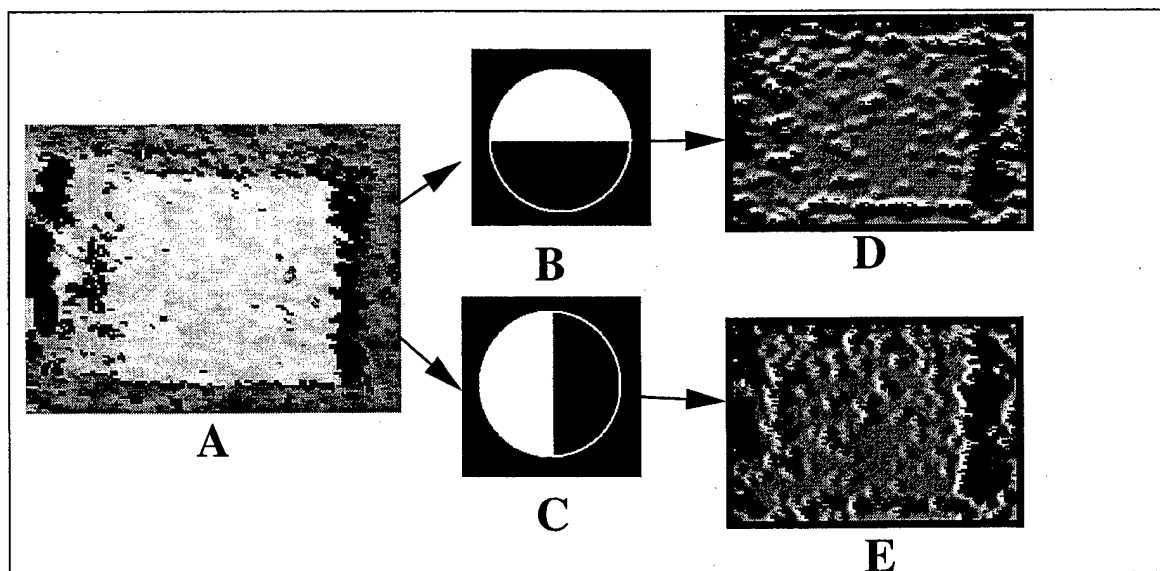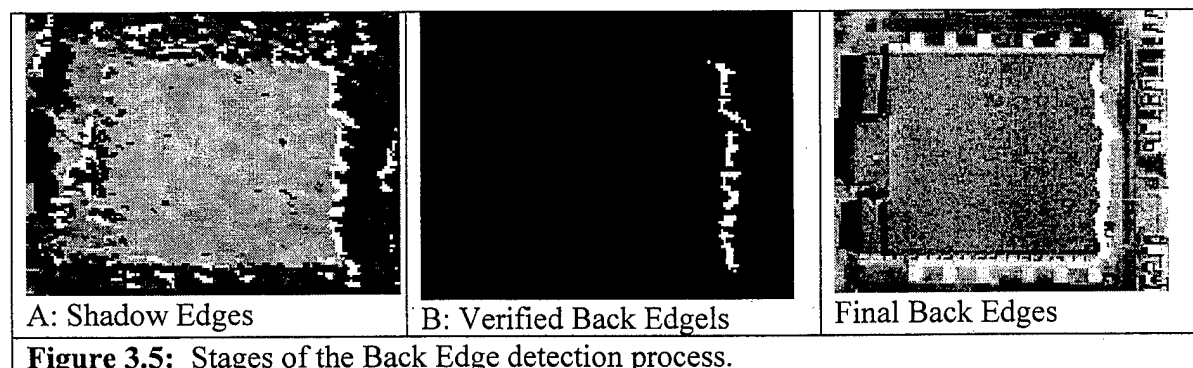

| Zero Degrees | 45 Degrees | Ninety Degrees | 350 Degrees |

Figure 3.3: Binary masks at varying orientations. These masks are used to determine if a point borders a shadowed region. The hypothesized border element separating shadow from rooftop is shown in grey.

Figure 3.4: A) A building's height map. B) Binary mask M_{270}. C) Binary mask M_0. D) Match scores resulting from the application of M_{270}. E) Match scores resulting from the application of M_0. A point's grey scale value is inversely proportional to its match score. As such, points receiving the best match scores will be the brightest in the D and E.
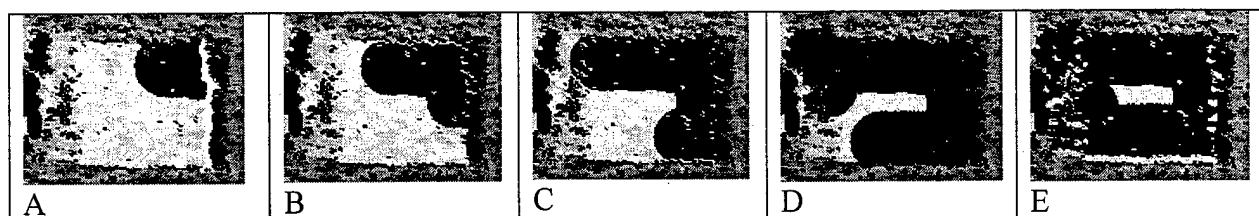

A: Shadow Edges   B: Verified Back Edgels   Final Back Edges

Figure 3.5: Stages of the Back Edge detection process.

Figure 3.6: Extracting the remainder of the building's boundary via region growing. The rooftop region grown so far is shown in black, while the back edge (Figure 3.5, far right) from whence it began is shown in white. The region's growth progresses panel A to panel E.
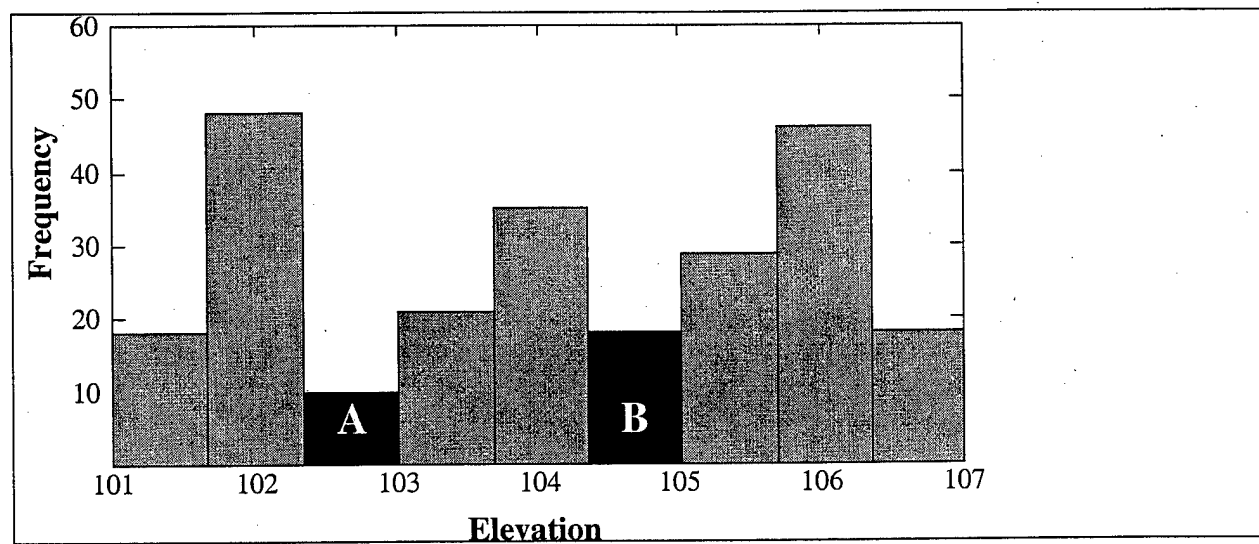

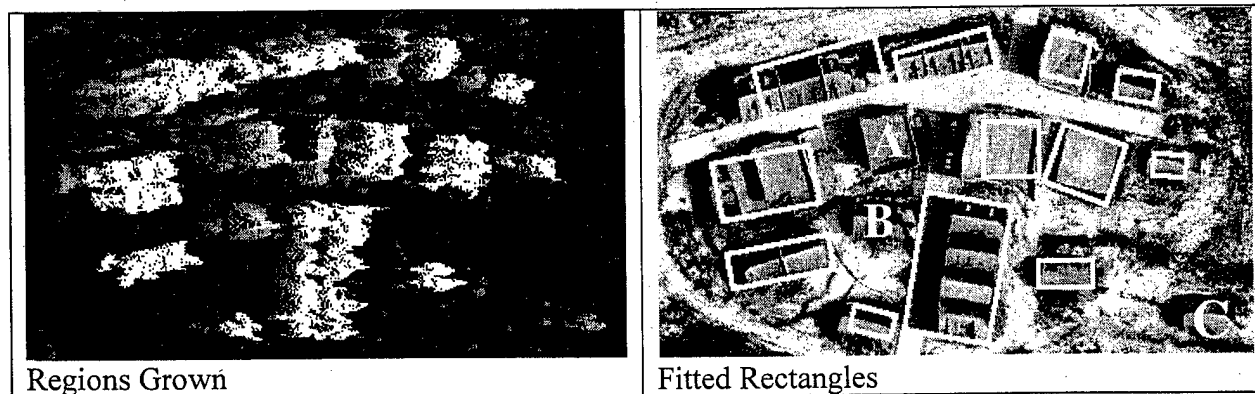**Figure 3.7:** A local elevation histogram used in determining the new classification threshold.

Regions Grown | Fitted Rectangles

Figure 3.8: Buildings extracted from the MOUT DEM. Buildings A, B, and C were not detected.
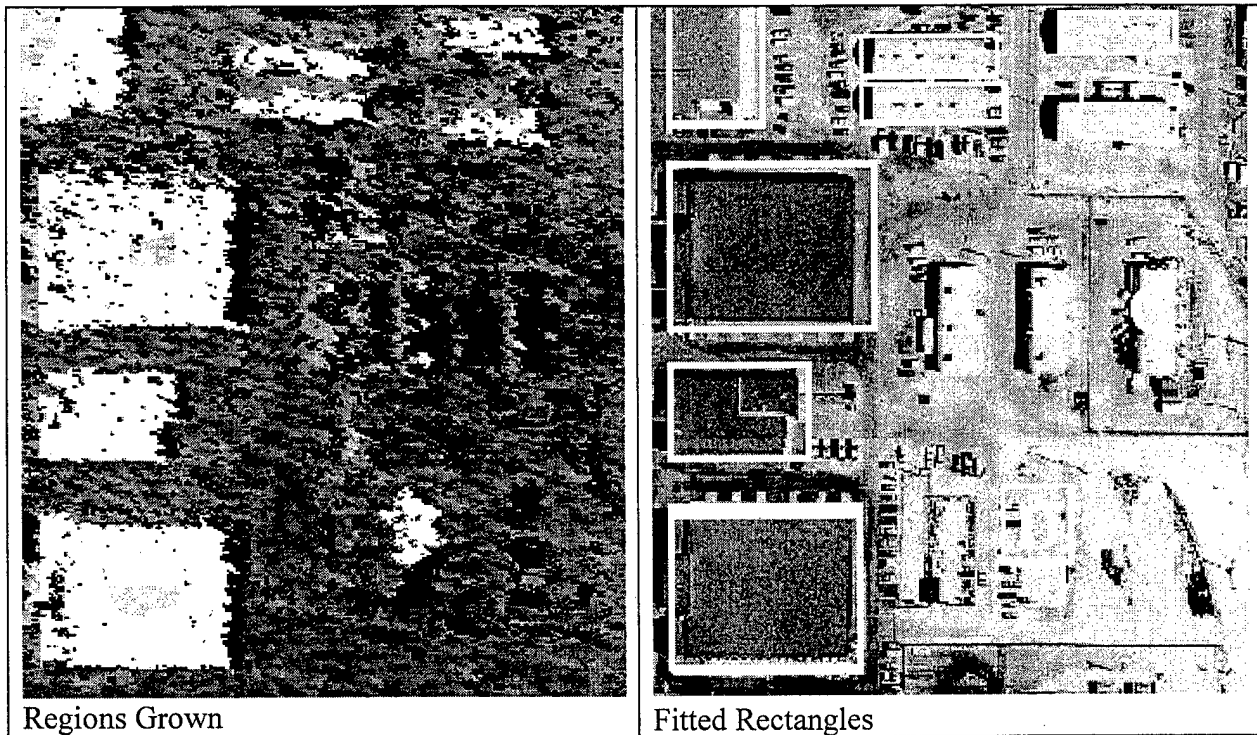
Regions Grown | Fitted Rectangles

Figure 3.9: Buildings extracted from the Kirtland AFB scene. Buildings A, B, and C were not detected. Building D was a false positive.
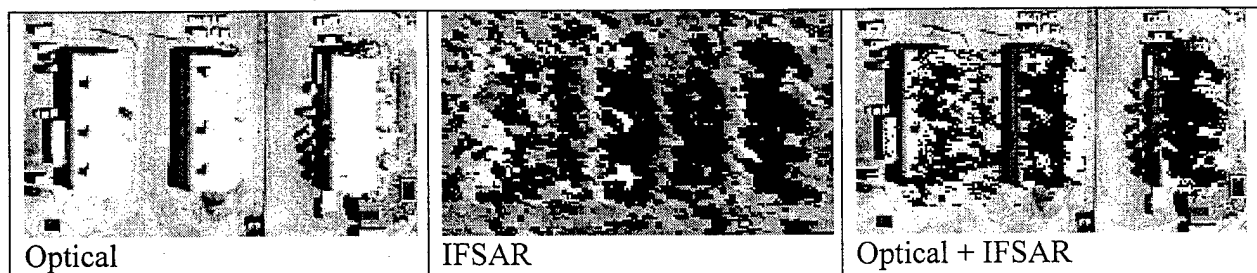

Optical | IFSAR | Optical + IFSAR

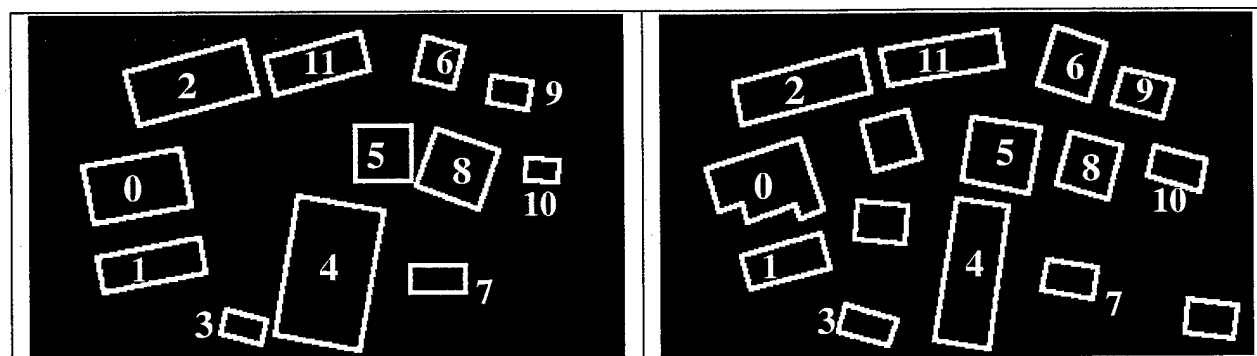**Figure 3.10**: Buildings dropped out of the Kirtland DEM.

Figure 3.11: Left: Boundaries extracted by the system. Right: Reference polygons hand-extracted from an orthorectified optical image.

53

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE September 2000	3. REPORT TYPE AND DATES COVERED Final Technical April 1997 - May 1999

4. TITLE AND SUBTITLE
Ascender II: Knowledge-Directed Image Understanding for Site Reconstruction

5. FUNDING NUMBERS
DACA76-97-K-0005

6. AUTHOR(S)
Allen Hanson, Edward Riseman, Howard Schultz

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Massachusetts
Computer Science Department
Lederle Graduate Research Center
Amherst, MA 01003-4610

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 N. Fairfax Drive, Arlington, VA 22203-1714

U.S. Army Topographic Engineering Center
7701 Telegraph Road, Alexandria, VA 22315-3864

19. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)* The Ascender II system was designed to perform three dimensional reconstruction of cultural objects (primarily buildings) from multiple aerial images. It is based on the premise that cooperative redundant reconstruction algorithms will succeed where individual algorithms fail and that a major task for a vision system is deciding which algorithm to apply to what data (and when). Control is based on Bayesian networks and utility theory is used to compute the marginal value of information for alternative operators and to select the one with the highest return. Two reconstruction algorithms are described that, along with other techniques, form the repertoire of algorithms. One algorithm reconstructs a 3-dimensional model of the scene using the differential geometry of scene surfaces to index into a set of model surfaces. A robust surface optimization converges on the model and parameters that most closely describe the data. After the best-fit surface has been determined, an outlier analysis phase searches for substructures that are recursively processed. The second algorithm recovers geometric structure from SAR and IFSAR data. The presence of noise, missing data, and poorly understood radar artifacts in such images necessitates the use of robust and context-sensitive techniques. The algorithm expoits knowledge about the geometric structure of buildings and how this geometry interacts with the sensor.

14. SUBJECT TERMS
SAR/IFSAR, Bayes Nets, building reconstruction, site model construction, image understanding, spatial modelling, aerial image analysis, model-based reasoning

15. NUMBER OF PAGES
59

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED